A NEW APPROACH TO ARABIC SIGN LANGUAGE RECOGNITION SYSTEM

BY

SALIHU OLADIMEJI ALIYU

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

MAY, 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **SALIHU OLADIMEMJI ALIYU** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING DEPARTMENT**.

Thesis Committee

M. Mohandes

Dr. Mohamed Mohandes (Adviser)

Dr. Mohamed Deriche (Co-adviser)

azzedinezerguine

Dr. Azzedine Zerguine (Member)

Dr. Ali Al-Shaikhi (Member)

Dr. Adil Balghonaim (Member)

Dr. Ali Ahmad Ali-Shaikhi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

28/5/15

Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

This thesis work is dedicated to Almighty Allah for His infinite mercy on me and my family

ACKNOWLEDGMENTS

In the name of Allah, the most Merciful, the most Gracious. All praise is due to Allah; we praise Him, seek His help, and ask for His forgiveness. The peace and mercy of Allah (SWT) be upon the holy prophet Mohammed (SAW), his household and companions.

First, I give thanks to Almighty Allah for which with His mercy, good works are completed. Secondly, special thanks to my wife, Fatimot Olabisi Salihu, for her contributions, patience and perseverance throughout the cause of this project, my parent for all their supports and care since the day I was born until this moment. May Allah have mercy on them. Thirdly, I would like to acknowledge King Fahd University of Petroleum and Minerals for the opportunity given to me and their supports throughout my stay in the Kingdom.

The author would like to appreciate the support of Dean of Scientific Research for funding this project. I would like to express my deep appreciation to my thesis advisor Dr. Mohamed Mohandes, and my thesis co-advisor Dr. Mohamed Deriche for their maximum support towards the completion of this project. My profound gratitude goes to my thesis committee members Dr. Azzedine Zerguine, Dr. Ali Al-Shaikhi, and Dr. Adil Balghonaim for taking time to review and contribute to *the success of this thesis.*

The author would also like to acknowledge Mr Uthman, the deaf signer, for taking time to provide real life data of Arabic Sign Language. My acknowledgment also goes to all faculty member, who taught me in KFUPM, for all the knowledge impacted on me. Thanks to Mr Maajid and Mr AL-Zamil, Thamer Najem for their effort and role in trying to bring a hearing impaired person to perform the sign language. The contributions of Mohammad Ali Qureshi and Toluwani Oluwatobi Daramola in making the thesis report look good are also appreciated. Many thanks to KFUPM community especially Nigeria community in KFUPM for making my stay in Saudi Arabia a wonderful one. Finally, my profound appreciation goes to the management Federal University of Technology, Minna, Niger State, Nigeria for the study fellowship given to me to embark on this study. To all of you who support this success, I say thank you and the reward of good is good.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	**Description**
LDA	Linear Discriminant Analysis
SLR	Sign Language Recognition
MLP	Multilayer Perceptron
NBC	Naive Bayes Classifier
ArSLR	Arabic Sign Language Recognition
LMC	Leap Motion Controller
ArSL	Arabic Sign Language
MK	Microsoft Kinect
IR	Infra Red
RGB	Red Green Blue
GMM	Gaussian Mixture Model
ML	Maximum Likelihood
EM	Expectation Maximization
IDE	Integrated Development Environment
PR	Pattern Recognition
MDC	Minimum Distance Classifier
PCA	Principal Component Analysis

KNN	K-Nearest Neighbor
DCT	Discrete Cosine Transform
PCNN	Pulse-Coupled Neural Network
GPU	Graphics Processing Unit
MMNN	Multi-level Multiplicative Neural Networks
SVM	Support Vector Machine
RBF	Radial Basis Function
ALI	Axis of Least Inertia
RF	Random Forest
DTW	Dynamic Time Warping
SDK	Software Development Kit
NB	Naive Bayes
ROC	Receiver Operating Characteristic
D-S	Dempster and Shafer

THESIS ABSTRACT

NAME: Salihu Oladimeji Aliyu

TITLE OF STUDY: A New Approach To Arabic Sign Language Recognition System

MAJOR FIELD: Electrical Engineering Department

DATE OF DEGREE: May, 2015

Sign language is important for facilitating communication between hearing impaired and the rest of the society. However, very few vocal people know sign language. Therefore, there is a need to develop systems to translate between spoken and sign languages automatically. The Arabic Sign Language (ArSL) has not witnessed research attention as other international sign languages. Two approaches have traditionally been used in the literature: image-based and glove-based systems. Glove-based systems require the user to wear electronic instrument while performing the signs. The glove includes a number of sensors detecting different hand and finger articulations. Image-based systems use camera(s) to acquire a sequence of images of the signer. Each of the two approaches has its own disadvantages. The glove-based method is not natural as the user must wear a cumbersome instrument *while the camera-based system requires specific background and environmental conditions to achieve high accuracy. In this thesis, we propose a new approach for ArSL recognition system which involves the use of the recently introduced device: Leap Motion Controller (LMC). Data was collected by using a native adult signer, for 100 isolated Arabic sign language words. Ten observations were collected for each of the signs to give a total of 1000 observations. On this data set, 70% was used for training and the rest for testing. A maximum recognition accuracy of 80.60% was achieved, on the test set, using a Gaussian Mixture Model (GMM) based classifier.*

ملخص الرسالة

الاسم الكامل: صالحو أولاديمجي عليو

عنوان الرسالة: منهاج جديد لنظام ترجمة لغة الإشارة العربية

التخصص: هندسة كهربائية

تاريخ الدرجة العلمية: مايو, ٢٠١٥م

لغة الإشارة هي مهمة لتسهيل التواصل بين ضعاف السمع وبقية أفراد المجتمع. ولكن قليل من الناس من يعرف لغة الإشارة. ولذلك، هناك حاجة إلى تطوير أنظمة للترجمة بين اللغة المنطوقة ولغة الإشارة تلقائيا. لم تشهد لغة الإشارة العربية (ArSL) اهتماما في الابحاث كغيرها من لغات الإشارة الاخرى. وقد جرت العادة على استخدام نهجين مختلفين للترجمة: النظم المستندة إلى تحليل الصور والنظم القائمة على القفازات الالكترونية. النظم التي تعتمد على القفازات تتطلب من المستخدم ارتداء أداة الإلكترونية أثناء تنفيذ الاشارات. تتضمن القفازات عدد من الحساسات الدقيقة للكشف عن حركة مختلف مفاصل اليد والأصابع. النظم القائمة على تحليل الصور تستخدم كاميرات للحصول على سلسلة من الصور لمنفذ الاشارات. كل من النهجين له سلبياته الخاصة. طريقة القفاز تجبر المستخدم على ارتداء أداة مرهقة مما يجعل الحركة غير طبيعية بينما نظام الكاميرات يتطلب خلفية وظروف بيئية محددة لتحقيق درجة عالية من الدقة. في هذه الأطروحة، نقترح نهجا جديدا لنظام التعرف على لغة الاشارات العربية يستخدم جهاز تحكم تم تطويره مؤخرا يدعى Leap Motion Controller (LMC). وقد تم جمع بيانات مائة مفردة من اشخاص صم ممن لديهم خبرة بلغة الاشارة العربية. وقد تم جمع عشرة نماذج مختلفة من كل مفردة سبعة نماذج استخدمت لتطوير النظام والثلاثة الباقية للتقييم. وفي نماذج التقييم تم التوصل إلى دقة تعرف بحدود ٨٠,٦٠٪ باستخدام Gaussian Mixture Model (GMM).

CHAPTER 1

INTRODUCTION

1.1 Background

Sign language is the natural means of communication between the hearing impaired and the rest of the society. Statistics show that over 5% of the world populations are hearing impaired [1]. The problem is that, very few vocal people understand sign language. Hence, the need to develop systems capable of automatically translating sign languages into words and sentences is becoming a necessity. In recent years, sign language recognition systems for American, British, Indian, Chinese, Turkish, and many international sign languages have received much attention as compared to the Arabic sign language. Therefore, in this work, we will be focusing on the Arabic sign language recognition system. Up to date, most developed systems for sign language recognition fall under one of two main categories: glove-based and image-based approaches.

The glove-based approach requires signers to wear an electronic sensor glove.

The sensors track and detect hands and fingers motion. The drawback of this approach is that the signer has to wear a cumbersome instrument while performing the signs [2]. Image-based approach uses image processing techniques to detect and track hands and fingers, as well as facial expressions of the signer. A disadvantage of this approach is that the segmentation of the hands and fingers requires extensive computations. The signer may be required to wear colored gloves to simplify the segmentation process. This approach is easier to the signer, however, some restrictions on background and lighting may be needed for better recognition accuracy [2].

In this thesis, we propose a completely new approach that eases the restrictions and constraints of the two currently available approaches. In particular, we propose to use the recently introduced Leap Motion Controller (LMC) [3]. The LMC detects and tracks hand and fingers motion while the sign is being performed. It was introduced as means of interactivity with computers using natural movement of hand and fingers for electronic games. It is finding application in wide areas besides gaming. However, it has not been used for Arabic sign language recognition before. Our first attempt of using the device for ArSLR at alphabet level has been presented in [4, 5]. In this thesis, we extend the usage for recognition of isolated words. We propose to use the LMC to acquire data for the hand and fingers motion while the sign is performed. The LMC has been proven to have 0.7mm precision with regard to gesture-based user interface [6]. Considering its high tracking precision, we propose to use the device as a backbone for Arabic sign language recognition.

Unlike Micrsoft Kinect (MK) device which can detect motion in its active range and provide information such as RGB, depth, and skeleton images among others, the LMC focuses on hand and fingers tracking, and provides discrete data of the object detected within its coverage. Similarly, complex computation are not needed to extract features from the LMC as in the case of the MK device. Therefore, we have used the device to collect data for the Arabic sign language alphabets and 100 signs.

In addition to data acquisition, the proposed system includes a preprocessing stage, a feature extraction stage, and a classification stage. Data collected from signs performed by an adult native signer is used in this work. The developed system relieves the signers from wearing cumbersome gloves and remove the background and lighting constraints which are limitations of the current approaches.

1.2 Problem Statement

Statistics shows that over 5% of the world population is hearing impaired [1]. To facilitate the communication between the hearing impaired and the rest of society, the sign language is used. The problem is that most vocal people do not understand sign language, hence, the need to develop electronic systems capable of translating sign languages into text or spoken language. A typical advantage of such system will be to install it in public places to aid communication. Several systems have been proposed in the literature, though very few in the area of

Arabic Sign Language. The requirement of constant background lighting and the wearing of cumbersomeness sensor-glove are among problems with current approaches, hence, hindering its user acceptability. Therefore, the problem lies in developing an Arabic Sign Language recognition system which will not require the user to use cumbersome sensor instrument and will remove some constrains in background lighting.

1.2.1 Thesis Objectives

The objectives of this thesis are:

- To develop a new approach to ArSLR that uses least restriction compared to current methods.
- To test the proposed system over a medium size vocabulary set consisting of 100 isolated word level signs.
- To investigate the most effective position for placing one or more LMCs and method for combining data from LMCs.

1.3 Major Contributions

The major contributions of this research work include the following:

- Data collection of 10 samples each of 100 Arabic Sign Language words performed by native deaf signer in the coverage area of the Leap Motion Controller.

- Development of algorithm for sign recognition from the data sets. Recognition stage involves training using 70% of the data set and testing stages using the rest of the data.
- Evaluating the performance of combining the data of two LMCs.
- Testing the developed setup on ArSL alphabet and isolated words.

1.4 Organization of Thesis

The rest of this thesis is organized as follows: In Chapter 2, we present a review of different approaches and techniques which have been used in the literature for ArSLR. First, we present a review on ArSLR using the image-based approach, and using the sensor-based approach. This is followed by a review of Sign Language Recognition (SLR) systems using LMC, MK devices, and finally, a review of different techniques for multi-classifier fusion is presented. In Chapter 3, we present the experimental setup and methodologies used for ArSLR. In Chapter 4, experimental results and discussions are presented in details. The results are presented for several classifiers and various scenarios considered in this thesis. We also present an alternative setup using the MK device. In Chapter 5, conclusion of the thesis is given with some potential future research directions.

CHAPTER 2

LITERATURE REVIEW

The image-based and the data glove have been the two traditional approaches used in the literature. The first category requires the use of camera(s) to capture signs performed by the signer. This is followed by segmenting the acquired images, feature extraction etc. This approach has a drawback in that specific camera and constant environmental background settings are required to achieve reasonable accuracy. The second approach requires the signers to wear a cumbersome sensor glove or a colored glove. The wearing of the color glove simplifies the task of hand and finger segmentation. However, the drawback of this approach is that the signer has to wear the sensors hardware along with the glove while performing the signs [7].

In the following sections, a brief review of previous work is presented. Other reviews presented in this chapter include sign language using the LMC, MK device and review of classifier combination techniques.

2.1 Image-Based Sign Language Recognition System

Traditionally, there are three vocabulary levels of ArSLR systems: alphabets, isolated words, and sentence level recognition. A typical image-based recognition system consists of 5 stages: image acquisition, pre-processing, segmentation, feature extraction, and classification. In sections 2.1.1 and 2.1.2, a review on alphabet and isolated words recognition systems for Arabic sign languages is presented.

2.1.1 Alphabets Sign Recognition

Under this scenario, the signer performs each letter separately. Mostly, letters are represented by a static posture and the vocabulary size is limited. In this section, several methods for image-based Arabic sign language alphabet recognition are discussed. The alphabets used for Arabic sign language are displayed in Figure 2.1.

Figure 2.1: Arabic Sign Language aphabets

Even though the Arabic alphabet only consists of 28 letters, the Arabic sign language uses 39 signs. The eleven additional signs represent basic signs combining two letters. For example, the two letters ”ال” are quite common in Arabic (similar to the article ”the” in English). Therefore, most literature on ArSLR uses these basic 39 signs.

In [8], Mohandes introduced an automatic recognition of the Arabic sign language letters. For feature extraction, Hu’s moments are used. For classification, the moment invariants are fed to support vector machines. A correct recognition rate of 87% was achieved. Al-Jarrah and Halawani [9] developed a neuro-fuzzy system. The main steps of the system include: image acquisition, filtering, segmentation, hand outline detection followed by feature extraction. Bare hands were considered in the experiments achieving a recognition accuracy of 93.6%. In [10], Al-Rousan and Hussain built an adaptive neuro-fuzzy inference system for alphabet sign recognition. A colored glove was used to simplify segmentation and geometric features were extracted from the hand region. The achieved recognition accuracy was 95.5%.

Assaleh and Al-Rousan [11] used a polynomial classifier to recognize alphabet signs. A glove with 6 different colors was used: 5 for fingertips and one for the wrist region. Different geometric measures such as lengths and angles were used as features. A recognition rate of about 93.4% was achieved on a database of more than 200 samples representing 42 gestures. In [12], Maraqa and Abu-Zaiter used recurrent neural networks for alphabet recognition. A database of 900 samples, covering 30 gestures performed by 2 signers, was used in their experiments. Colored gloves similar to the ones in [11] were used in their experiments. The Elman network achieved an accuracy rate of 89.7% while a fully recurrent network improved the accuracy to 95.1%.

In [13], El-Bendary et al. developed a sign language recognition system for the ArSL alphabets achieving an accuracy of 91.3%. In their system, images of bare hands were processed. The input to the system is a set of features extracted from a video of signs and the output is simple text. For each frame, the hand outline is first extracted. Using a centroid point, the distances to the outline of the hand covering 180 degrees are extracted as a 50 dimensional feature vector. These features are rotation, scale, and translation invariant. In the feature segmentation stage, they assumed a small pause between letters. Such pauses are used to separate the letter numbers and the related video frames. At the recognition stage, a multilayer perceptron (MLP) neural network and a minimum distance classifier (MDC) were used.

Hemayed and Hassanien [14] discussed an Arabic sign language alphabet recognition system which converts signs into voice. The technique is much closer to real life setup however; recognition is not performed in real time. The system focuses on static and simple moving gestures. The inputs are color images of the gestures. To extract the skin blobs, the Luma, blue-difference and red-difference Chroma components (YCbCr) space was used. The Prewitt edge detector is used to extract the hand shape. To convert the image area into feature vectors, Principal Component Analysis (PCA) is used with a K-Nearest Neighbor (KNN) Algorithm in the classification stage. Naoum et al. [15] developed an image-based sign language alphabet recognition system with an accuracy of 50% for naked hand, 75% for hand with a red glove, 65% for hand with a black glove and 80% for hand with a white glove. The system starts by finding histograms of the images. Profiles extracted from such histograms are then used as input to a KNN classifier.

Arabic alphabet signs recognition is the simplest among all image-based ArSLR approaches as the vocabulary size is limited and the signs are represented with mostly static images. Such systems achieve high recognition rates of over 90%. Note, however, alphabet signs are not commonly used in daily practice. Their use is limited to finger spelling of words without specific signs like proper names. Much of the current research efforts have been put into developing systems that focus on isolated words or even continuous sign recognition.

2.1.2 Isolated Signs Recognition

Contrary to alphabet sign recognition, word sign recognition techniques analyze a sequence of images representing the entire sign, as shown in Figure 2.2, [16].

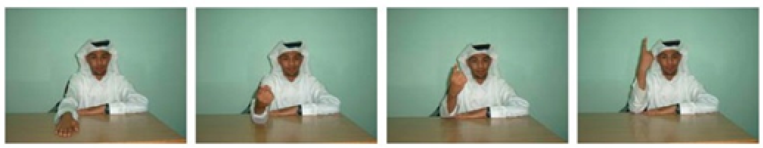


Figure 2.2: The image sequence of "1"

In [17], Mohandes and Deriche used a Hidden Markov Model (HMM) to identify isolated Arabic signs from images. They used a dataset consisting of 500 samples representing 50 signs. A Gaussian skin color model was used to find the signer's face which is then taken as a reference for the hands movement. Two colored gloves (orange and yellow) were used for the right and left hands for ease of hand region segmentation as shown in Figure 2.3. A simple region growing technique is used for hands segmentation. The recognition rate achieved over 50 signs was 98%. In [18], the same authors extended the work to cover a dataset of 300 signs achieving a recognition accuracy of 95%.

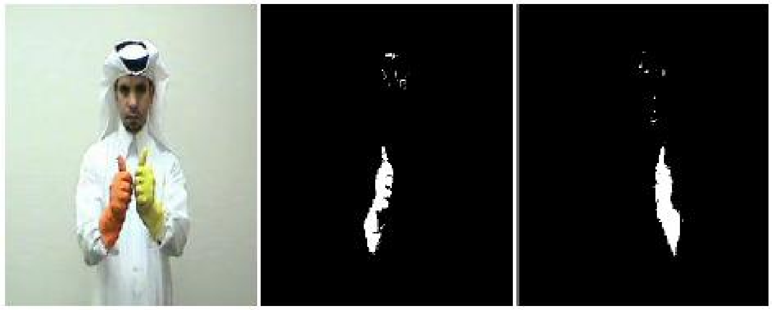


Figure 2.3: The extracted right and left hand regions

Shanableh et al. developed a signer-independent system for isolated Arabic signs [19]. They used segmented images of the hands extracted from colored gloves. For feature extraction, they used zonal DCT coefficients, while KNN was used for classification. The authors achieved a classification rate of 87% over a vocabulary size of 23 signs. The same authors extended their work using HMM-based classification [20, 21]. They introduced new video-based features where motion is taken into account. The system achieved a recognition accuracy of about 95%.

In [22], Youssif et al. developed an ArSLR system for isolated signs using HMM. The regions of the palm and the fingers were modeled as ellipses and circles. They used a limited vocabulary size of 20 signs. With only 8 features they were able to achieve an accuracy of 82.2% under glove free signer independent mode. Zaki and Shaheen [23] presented a combination of appearance based features. Kurtosis position was used to identify the articulation location, while PCA was used to represent the hand region, and they used a motion code chain to represent the hand movement. With a database of 50 signs, the system achieved a recognition accuracy of about 90%.

In [24], Samir and Aboul-Ela proposed a semantic-oriented approach. Natural language processing rules are used to detect and correct errors from the classification stage. The proposed approach was shown to enhance recognition accuracy of ArSLR by around 20%. In [25], Elons et al. used a Pulse-Coupled Neural Network (PCNN) for image features generation from two different viewing angles. The features were evaluated using a fitness function to obtain a weighting factor for each camera. The features derived from the two images were used to obtain 3D optimized features. The dataset used in the experiment contains 50 isolated words and the achieved recognition accuracy was 96% for pose-invariant restrictions with a tolerance of up to 90 degrees of freedom.

Elons in [26], proposed a Graphics Processing Unit (GPU) for real-time recognition of Arabic Sign Language using Multi-level multiplicative neural networks (MMNN). The system architecture depends on two layer of MMNN, where the first layer determines the number of hands used by the signer, while second layer performs the sign recognition. A maximum recognition rate of 83% was achieved on 200 signs. In [27], Al-Rousan et al. developed a system which was able to perform automatic translations of dynamic signs. The proposed hierarchical system divides signs into groups. For a given test sign, the group is first identified followed by the sign recognition within that group. Twenty three geometric features were used and tracked with an HMM classifier achieving a recognition accuracy of 70.5% for user-independent mode and 92.5% for user-dependent mode.

Isolated word sign language recognition is more practical; however, it is much more complex than alphabet recognition. More importantly, word recognition systems are required to deal with a sequence of images. The time component in analyzing such a sequence of images is very important. Note also that the vocabulary size for such systems can be very large. The challenge still remains in dealing with signs that are separated by certain pauses between signs. It is observed that the larger the vocabulary size, the less the accuracy becomes. For Arabic sign language, the size of the vocabulary needed for practical situations is still an open area for further research. In summary, the challenge for Arabic sign language recognition system is to develop signer independent systems that will deal with large vocabulary size suitable for practical deployment and will not require the signer to wear glove nor require specific background settings.

2.2 Sensor-Based Sign Language Recognition System

Sensor-based recognition methods process data acquired from gloves equipped with sensors. The PowerGlove [28], DataGlove [29, 30], and CyberGlove [31], have commonly been used for Arabic sign language recognition. These types of gloves are shown in Figure 2.4.

(a) Power glove

(b) Data glove

(c) Cyber glove

Figure 2.4: Types of glove

These gloves provide information on the position, rotation, movement, orientation of the hand, and more importantly, finger bending. A large number of features can be extracted from the data provided by the gloves. These features can be used with a proper classifier to recognize the performed sign. In [32, 33], Mohandes et al. used a cost effective off-the-shelf device to implement a robust ArSLR system. Statistical features are extracted from the acquired signals and used with an SVM classifier. With a database of 120 signs, the authors achieved recognition accuracy of over 90%.

In [34], Assaleh et al. developed a low complexity classification system. The glove used in their system had 5 bend sensors and a 3D accelerometer. From the acquired data, a number of statistical parameters were estimated. A regression technique was used to rank and select the most relevant features. The final list of selected features was used with a KNN classifier. With a database of 10 signs performed by 10 different signers, a recognition accuracy of 92.5% was achieved in signer-independent mode while this accuracy was 95.3% for signer-dependent scenario. Ritchings et al. developed a computer-based system using the DataGlove for teaching sign language [35]. Bend sensors and push button switches were used to acquire 17 signals. The focus of the system was on assessing the ability of trainees in replicating signs performed by an expert signer. The database used covered 65 signs performed by four professional signers (teachers). The trainees were able to duplicate the signs with an accuracy of 93%.

In [36], a first attempt of two-handed Arabic signs recognition was made. The database consists of 20 samples from each of 100 two-handed signs performed by two signers. Second order statistics from sub frames of the signs were used as features. The length of the feature vector is then reduced using PCA. For classification, the SVM was used achieving an accuracy of 99.6% with 100 signs. In [37], Mohandes and Deriche used the Dempster-Shafer theory of evidence to combine decisions from the CyberGlove, as shown in Figure 2.5, with 22 sensors and the hand tracking system.

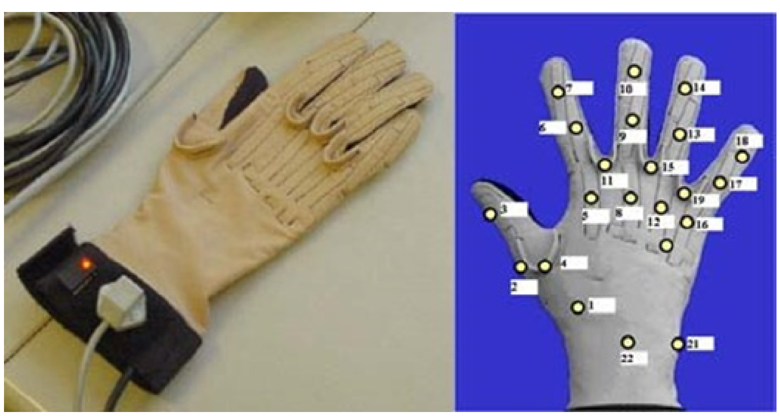


Figure 2.5: The CyberGlove system

The authors showed that the fusion at the decision level outperforms the traditional feature based fusion. They started with some basic experiments using the glove-based and the image-based systems independently. The image-based system achieved an accuracy of 84.7% while the glove-based system achieved an accuracy of 91.3%. The traditional feature based combination provided a maximum accuracy of 96.2% which was improved to 98.1% when fusion at the decision level is performed.

2.3 Sign Language Recognition using LMC

In [4], we introduced the first attempt of using the Leap motion controller for Arabic sign language recognition system. We started by evaluating the performance of the device on 28 ArSL alphabets. Since then, other authors have attempted to use the device for gesture recognition. In [38], the authors used Leap Motion Controller on 50 different dynamic Arabic language signs performed by two different signers. Temporal and spatial features were extracted from the Leap Motion data and fed to an MLP classifier achieving recognition accuracy up to 88%. Two sets of features were used: fingers position and fingers distances. Using fingers positions gave an accuracy of 82% on 50 signs while fingers distances achieved an accuracy of 88% on same data size.

In [39], the authors evaluated the suitability of the Leap Motion Controller for application in Australian Sign Language recognition system. There experiment revealed that, though the device can provides hand, finger movement tracking accurately, however, detection accuracy reduces when the hand is in position that obstruct the device view. A typical example occurs when the hand is placed perpendicular to the surface of the Leap device. The authors further conclude that the LMC can be used to recognize basic signs, but not suitable to recognize complex signs such as signs that involves face or body contact. Leap Motion Controller have been used in combination with Surface Electromyography (SEMG) to enhance gesture recognition rate [40].

In [41], Marin et al. proposed the used of Leap Motion Controller in combination with Microsoft Kinect for hand gesture recognition. Features based on fingertips positions and orientations were fed into a multi-class SVM classifier. In order to improve the recognition rate, another set of features was extracted from the Microsoft Kinect (MK) device and combined with the LMC. The complete feature set was obtained by concatenation of the two set of features ($V = [V_{LMC}, V_{MK}]$). Where V_{LMC} is the feature vector for LMC and V_{MK} is the feature vector for MK device. The set up was tested on American Manual Alphabet obtaining a recognition rate of 91.3%, from combination of the two devices, with multi-class SVM as classifier.

In [42], same authors used eight sets of features, four sets of features per device, on two separate classifiers: multi-class SVM and Random Forest (RF). The approach was tested on 10 different static gestures performed by 14 different people. Each gesture was repeated 10 times giving a total of 1400 samples. Different cases of combination of the different set of features from MK and LMC were experimented. A maximum accuracy of 96.5% was achieved with a feature dimension of 435 (consisting of all features from MK and LMC) using SVM and 94.7% with RF classifier. On applying feature selection strategies to reduce the feature dimension, obtained accuracy dropped. In [43], Karthick et al. proposed a system for transforming Indian Sign Language to text using the Leap Motion Controller. The system uses Dynamic Time Warping (DTW) and an Intelligent Sense (IS) algorithm to convert hand gestures into appropriate text.

The LMC is unlike the MK device which can gives RGB, and depth image of the signer. Since the introduction of the MK, the device has witnessed wide spread application in different fields of study. Application using the device has also extended to gesture recognition. In [44], Chai et al. proposed a sign language recognition and translating system using depth and color images obtained from the MK device. In their work, 3D motion trajectory of each sign language vocabulary was aligned and matched between probe and gallery to get the recognized result. They tested their approach on 239 Chinese SL words. Using two different ranking approaches, they achieved recognition rates of 83.51% and 96.32%.

In [45], Agarwal and Thakur presented a sign language recognition system which makes use of depth images that were captured using a Microsoft Kinect camera. Using computer vision algorithms, they developed a characteristic depth and motion profile for each sign language gesture. The feature matrix generated was trained using a multi-class SVM classifier and the final results were compared with existing techniques. Their work was based on recognizing Chinese

Sign Language gestures for digits $0 - 9$. Their experiments were conducted on two data sets, each consisting of 47 video sequences with individual gestures and sequences of multiple gestures. On the first data set, a recognition rate of 89.63% was achieved with linear kernel classifier and 92.32% with RBF neural network classifier. However, on the second data set, they achieved an accuracy of 77.59% with linear kernel classifier and 90.83% with RBF classifier.

In [46], Geetha et al. proposed a dynamic gesture recognition system using depth images obtained from MK device. They proposed a new trajectory based feature extraction method using the concept of Axis of Least Inertia (ALI) for global feature extraction. Other works where the MK device has been used include [45, 47, 48, 49, 50, 51, 52, 53]. Similar to the LMC, the MK device through its depth image can ease the issue of constant lighting background environment required in image-based. However, the LMC has less computational complexity, segmentation and the kind of feature extraction algorithms involve in MK are not required.

2.4 Review on Decision Fusion Techniques

Issues such as missing data, insufficient data sample, and curse of dimensionality etc, have led to the idea of decision combination from multiple sources. Decision fusion or combination can be done at three different level: sensor data level, feature and classifier decision level. On these three levels, several techniques have been proposed in the literature for fusion of classifier ensembles. In [54], the authors considered the problems and issue regarding classifier fusion. The authors were able to prove that despite advances in machine learning based on the concept of support vectors, the conventional approach to classifier designs such as feature selection, contextual classification and classifier fusion are still relevant to achieve a reliable PR system.

In [55], Michael et al proposed a method based on majority voting approach. The approach combines ensembles of classifiers using dynamic weighted consult-and-vote for incremental learning of new class. The approach was an improvement over a previously developed approach by the author, which suffers from inherent "out-voting" problem in learning a new class. Voting weights were determined by relative performance of each classifier on training data. In case a new class is introduced, the approach learns it by allowing individual classifiers to consult with each other to determine their voting weights for each of the test instance.

In multiple classifier fusion, individual classifiers either use the same representation of the input pattern or each uses its own representation of the input pattern [56].

In [57], Kittler et al developed a common theoretical framework for combining classifiers. In their work, they focus on ensembles which uses distinct pattern representation of the input pattern. However, in [56], both cases were considered. Starting from the Bayesian decision rule, the following combination rule were developed: max, min, median, and majority voting rule.

$$assign\ Z \longrightarrow w_j\ if\ P(w_j|\mathbf{x}_1, ..., \mathbf{x}_N) = \max_k P(w_k|\mathbf{x}_1, ..., \mathbf{x}_N), \qquad (2.1)$$

where Z is the pattern to be classified, w_j is the j^{th} class, and $\mathbf{x}_N$ is the feature vector of dimension N. Experimental comparison carried out on these rules shows that the sum rule outperformed other rules despite being developed under the most restrictive assumptions. Another classifier fusion approach that has gain widespread application is the Dempster-Shafer (D-S) theory of evidence developed by Dempster and Shafer. The theory introduced the system of beliefs in the output results which were not discussed in previous combination techniques [58], and it gives meaningful reason for combination results obtained. It is finding wide use in modeling uncertainty [37, 58, 59, 60]. The theory is based on three basic concepts: basic belief assignment, belief function and plausibility. In this work, we have used D-S theory, which is a classifier level combination, and feature level combination. More details on D-S theory is presented in chapter 3.

2.5 Summary

In this chapter, we presented a review of previous works in ArSLR under the two major approaches: Image based and Glove based. Previous works on alphabet and isolated word level recognition system were reviewed. We also presented review on emerging approaches which involve the use of LMC and MK device. Finally, we presented review on some techniques which have popularly been used for combination of classifier decision. In the next chapter, focus will be on our experimental set up and methodologies used in this thesis.

CHAPTER 3

METHODOLOGY AND IMPLEMENTATION

This chapter presents the main contributions of the thesis. First, we present our model for ArSLR using LMC(s), starting with single LMC model, followed by using a two LMCs model. As previously stated in chapter 2, research in ArSLR is divided into three categories: alphabet, isolated words and continuous sentence recognition. Major focus is on recognition of isolated words, with extension to sentences. The developed model in this thesis was tested on alphabet and isolated sign word recognition. Figure 3.1 shows the system block diagram. It involves the collection of data from the LMC(s), extraction of relevant features, training of the classifier algorithm and sign recognition. The block diagram represents the general idea of all the various setups discussed in this work. Sections 3.1 and 3.2 discuss the setup used for alphabet recognition using a single and two LMCs, while section 3.3 discusses the more general case of isolated word signs. We will show also the importance of using 2 LMCs as single-LMC system can lead to erroneous recognition given the detected between signs.

3.1 Alphabet Level Recognition using One LMC

The idea of using the LMC for ArSLR was first tested on 28 Arabic Sign Language alphabets as a proof of concept. The steps involved are summarized in the block diagram shown in Figure 3.1.

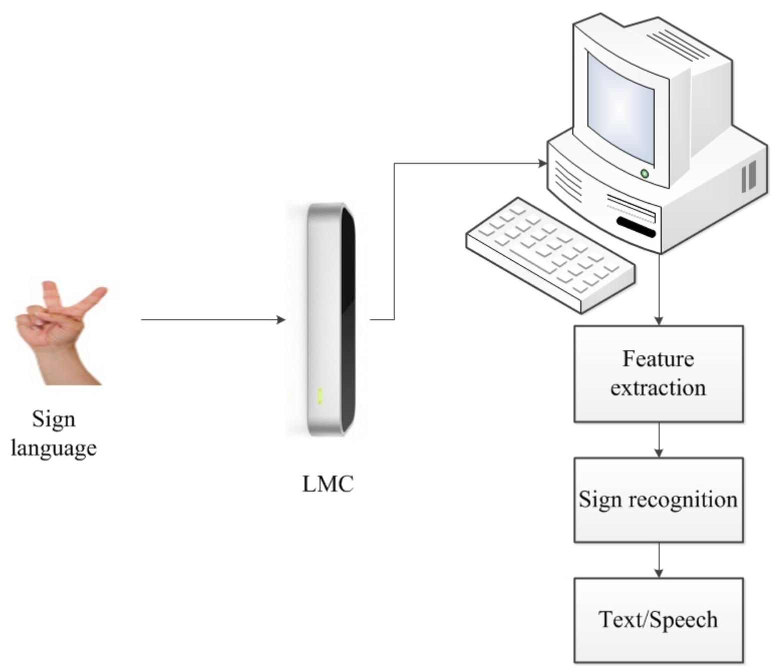
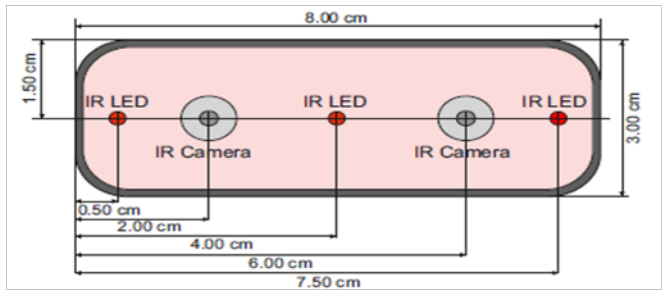


Figure 3.1: System block diagram

The LMC captures the hand motion as the sign is performed. Frames of data are collected from which discriminative features are extracted and used for sign recognition through machine learning techniques. In the following, each of the items in the block diagram is discussed in more details.

3.1.1 The Leap Motion Controller (LMC)

The LMC is an electronic device recently developed by Leap Motion Company [61]. The device detects and tracks hand motion, fingers and finger-like objects reporting discrete position, gestures and motion. It operates in a close proximity with a rate of 200 frames per second [3]. The LMC field of view is an inverted pyramid of about 8 cubic feet with center located on the device [62]. The device has functional range which increase from approximately 25 to 600 mm above it [3]. The device uses two high precision cameras and three infrared LEDs to capture information within its interaction range. However, it does not provide pictures or cloud data of detected images. Its driver software processes the acquired data, extracts position information using complex mathematics [62]. Figure 3.2 shows schematic view of the LMC as well as a true picture of the device [6].



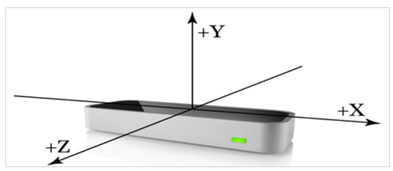
(a) Schematic view of LMC

(b) The LMC

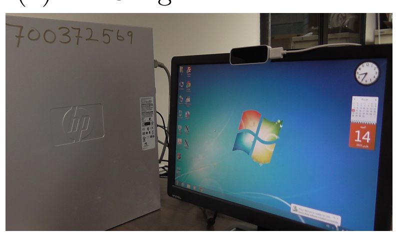
Figure 3.2: The LMC and its schematic

3.1.2 The LMC Coordinate System

The Leap Motion system uses a right-handed Cartesian coordinate system, with values reported in real-world millimeters. The center of the LMC serves as its origin. The horizontal plane is the x-z plane, the x-axis runs parallel to the long edge of the device, while the y-axis is vertical, with positive values increasing upwards. Values of z-axis are positive and increases away from the computer screen as shown in Figure 3.3) [3].



(a) LMC right-handed coordinate system.



(b) LMC setup.

Figure 3.3: LMC coordinate and PC setup

3.1.3 Motion Tracking

As the device detects hands and fingers in its field of view, it provides data updates as frames of data. Each of these frames contains lists of the basic tracking data, such as hands, fingers, as well as recognized gestures (if detected) and factors describing the overall motion in its view. In the event of the LMC detecting a hand(s), finger(s), or gesture(s), its driver software assigns to the detected object a unique ID tag. Provided the object remains within the LMC view, the tag ID remains the same. In a situation where tracking is lost and regained, the software may assign for it a new ID. Java program was written, using the NetBeans IDE, to collect the motion tracking data. Figure 3.4 shows frames of motion tracking data when the hand is being tracked.

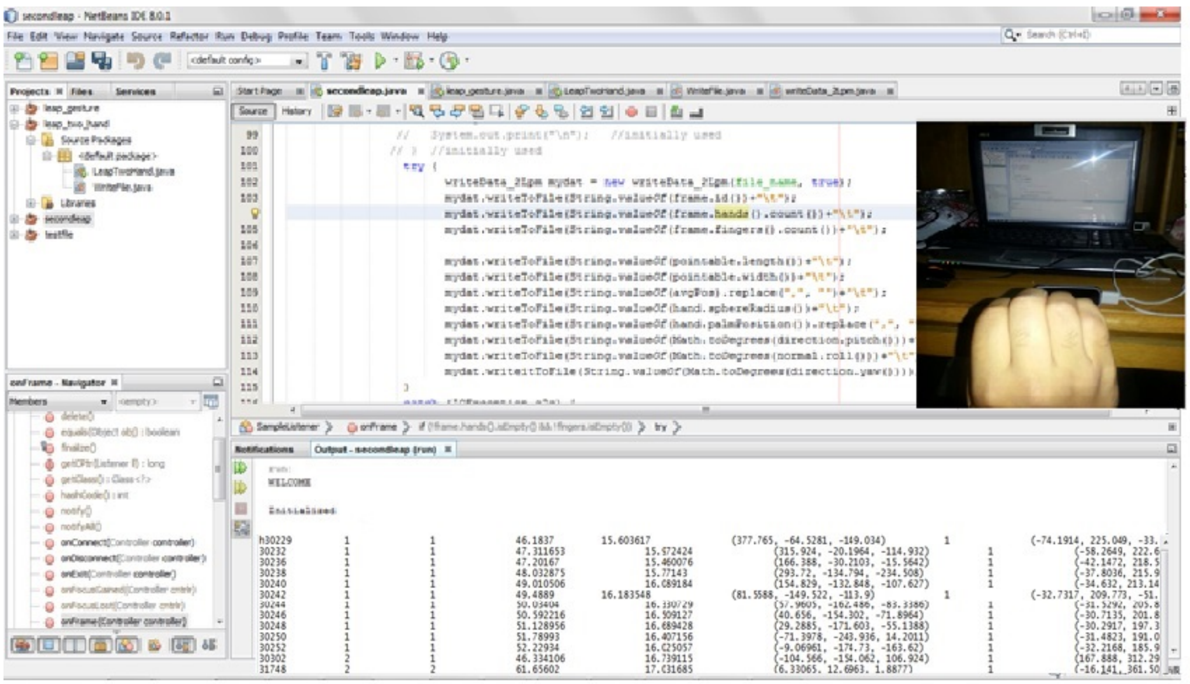


Figure 3.4: Motion tracking data with LMC

Figure 3.4 shows the interface for collecting the hand motion data. As the sign is performed, the hand data are displayed below as can be seen from the figure. In addition, the data is automatically saved on the PC for further analysis.

3.1.4 Data Collection for One LMC Setup

Our initial goal was to use the LMC with a machine learning algorithm to identify hand gestures representing static images of letters used for Arabic language. With the LMC software development kit (SDK) installed on the PC, the device was connected to the PC and the Java program was executed to collect the hand data from the LMC. The data collection was performed for the Arabic sign language letters (ا - ي), shown in Figure 3.5, using one LMC setup. Note that some of the references also considered hamza "ء" as a letter for the alphabets, hence 29 letters are considered.

Figure 3.5: Arabic sign letters (28 signs)

A single sample or observation is when each of the considered sign is performed once in front of the LMC device. Each sample contains several frames depending on how long the signer keep the hand in position. In our case, we have considered 10 frames per sample.

3.1.5 Feature Extraction and Analysis

As previously discussed, the LMC returns data in frames. Each frames consists of different geometric parameters describing the motion of object in the LMC view. Our data collection stage, based on the LMC SDK, returns twenty-three (23) geometric parameters describing the hand motion in the field of view of the LMC. In order to focus on relevant features for the classification stage, we carried out a simple statistical analysis of these parameters. We estimated the mean of each parameter across the 10 frames of each sample for the individual classes (signs).

For example, the mean value measure for the signs of letters أ and ب are plotted against sample number in Figures 3.6 and 3.7 respectively. The parameters found not discriminative enough were ignored. For a parameter to have strong discriminative power, it should have small within-class variance and large between-class variance. The discriminative parameters were extracted as features for the training and classification stage, while others were ignored. Examples of such parameters (or characteristics) include the frame Id, the numbers of hand, the tip velocity of the fingers etc. The frame Id is not an attribute of the hand, it's just a tracking tag attached to each frame of data, while the numbers of hand is constant in the case of sign letter recognition. Based on this initial analysis, the following features, listed in the Table 3.1 were extracted. In total, 12 features were extracted from each frame of data. See Appendix for image description of these features.

Table 3.1: List of Features

Features	Feature Name
1	Finger Length
2	Finger width
3	Average tip position along x-axis
4	Average tip position along y-axis
5	Average tip position along z-axis
6	Hand sphere radius
7	Palm position along x-axis
8	Palm position along y-axis
9	Palm position along z-axis
10	Hand pitch
11	Hand roll
12	Hand yaw

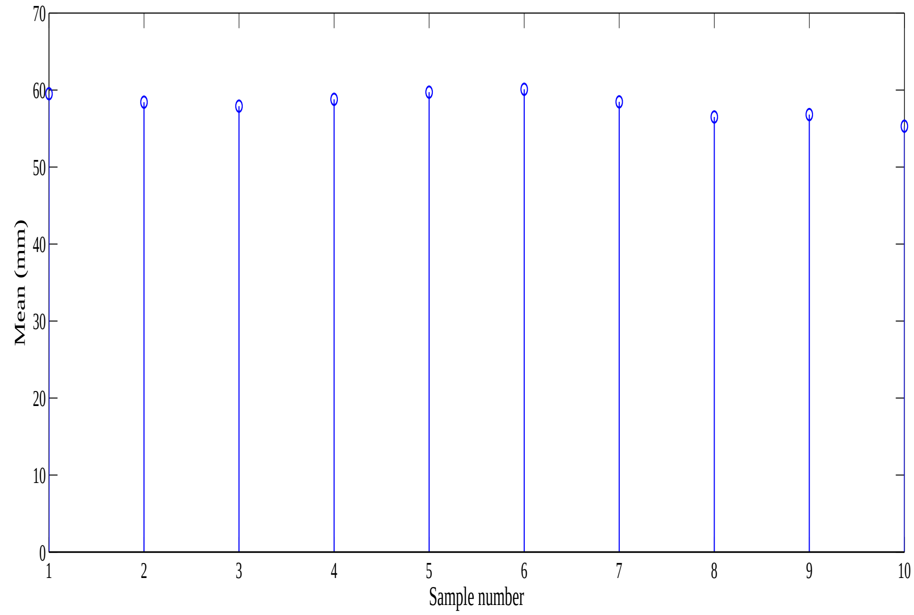


Figure 3.6: The mean value (across the 10 frames) of the feature finger length for each of the 10 samples of letter أ

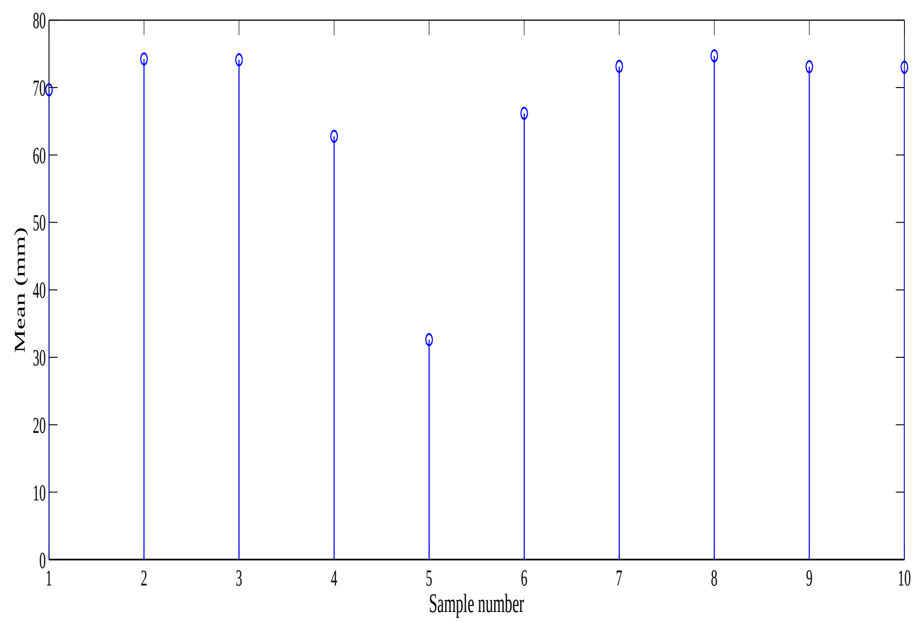


Figure 3.7: The mean value (across the 10 frames) of the feature finger length for each of the 10 samples of letter ب

The figures show that there are variations on the values of each feature related to the same letter, though the variation is small. This is due to the fact that usually people do not repeat a sign exactly the same manner making the classification process a challenging task. A good feature should have small within class variance and large between class variance. Similar to the finger length feature, we estimated the mean values across the 10 frames of the remaining 11 features. Similar figures to Figures 3.6 and 3.7 were obtained for other features. The mean of each of the extracted features across the 100 frames (10 frames from each of 10 samples) of letters were also obtained and displayed in Figures 3.8 and 3.9. A typical example on the discriminative power of the features is shown in Figure 3.10, using the hand roll feature for two class (letter أ and ب). From the decision boundary shown in the figure, we can see how discriminative the roll feature can be for two different classes (signs). It is obvious that classification could be more complicated in the case of the entire 28 alphabets. Each tap of Figures 3.8 and 3.9 represents a feature according to Table 3.1. Also, the unit of the mean-axis of Figures 3.6, 3.7, 3.8 and 3.9, is in mm.

3.1.6 Training and Classification

Based on the features discussed above, we compared the performance of two classifiers, namely, the Multilayer Perceptron (MLP) Neural Network (NN) classifier and the Naive Bayes Classifier (NBC). Generally, no single machine learning algorithm is appropriate to all PR problems. Since the dataset used here is new, we

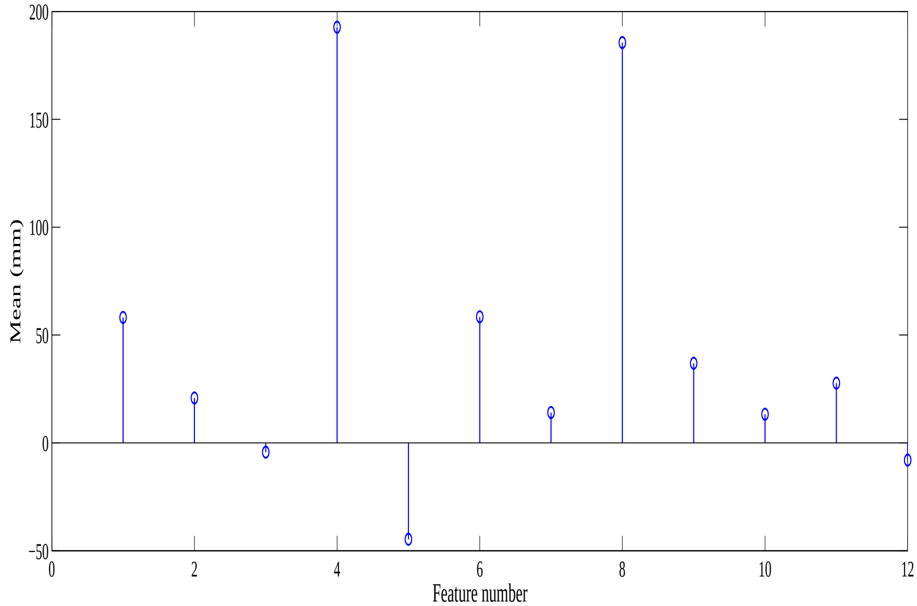


Figure 3.8: The mean value of each of the 12 features for letter أ

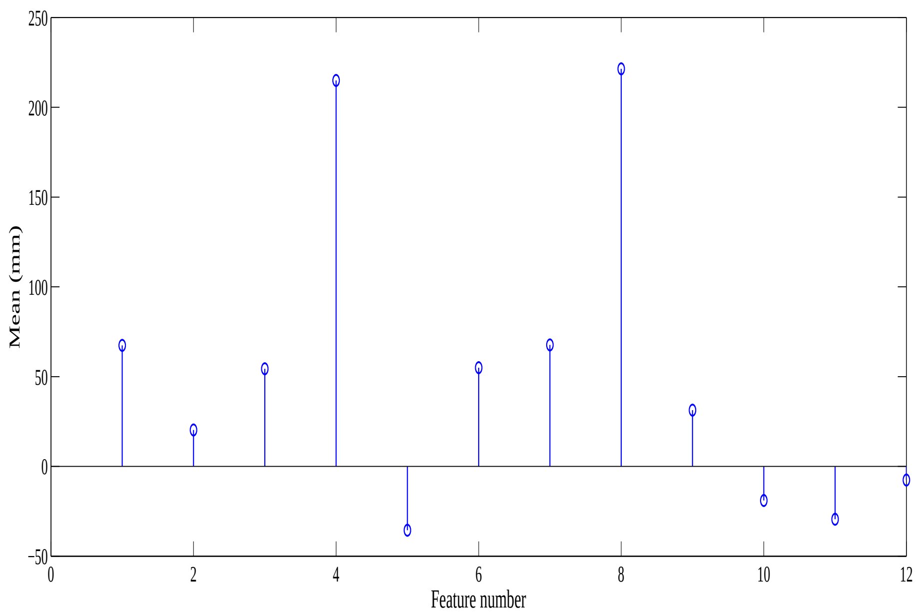


Figure 3.9: The mean value of each of the 12 features for letter ب

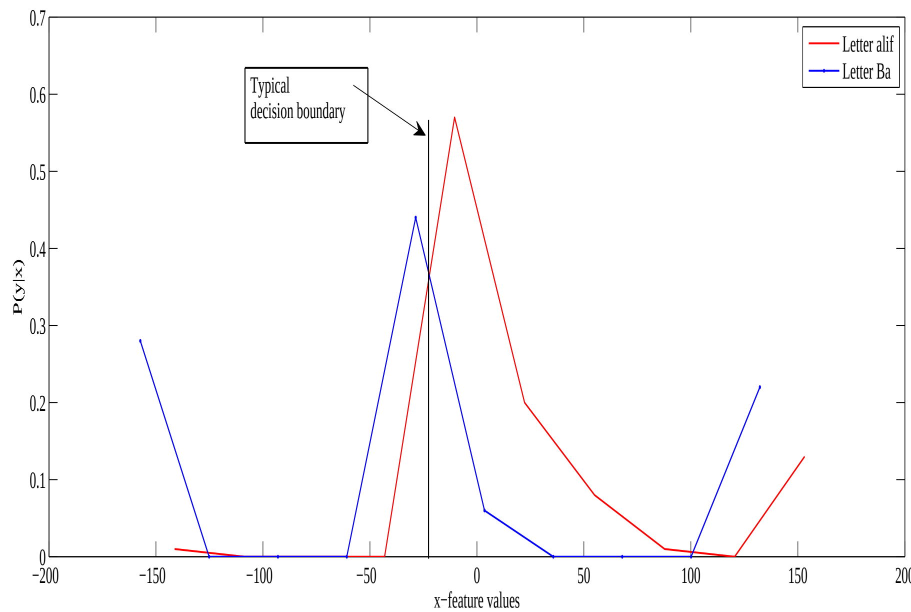


Figure 3.10: Hand roll feature for two classes (signs)

therefore propose to use a simple probabilistic classifier (NBC) as a baseline and compared it against a neural network classifier (MLP). MLP and Radial Basis Function (RBF) have much in common than most other NN learning algorithms. The basic difference is in the way hidden units are combined from previous layers in the network. In addition, MLP uses supervised learning approach for training, while RBF uses unsupervised learning. Since the task here is completely supervised approach, we have chosen to use the MLP. The MLP and NBC are briefly described below.

Multilayer Perceptron (MLP) Classifier

The MLP is an Artificial Neural Networks (ANN) based classifier. Our interest in using ANN was largely motivated by the main advantage of such systems in being able to mimic natural intelligence in learning from experience [63]. ANNs learn from examples by constructing an input-output mapping without explicit derivation of the model equations. ANNs have been used in a broad range of applications including: pattern classification, function approximation, optimization, prediction and automatic control, among others [64, 65]. The basic structure of an artificial neural network consists of many interconnected identical simple processing units called neurons as shown in Figure 3.11.

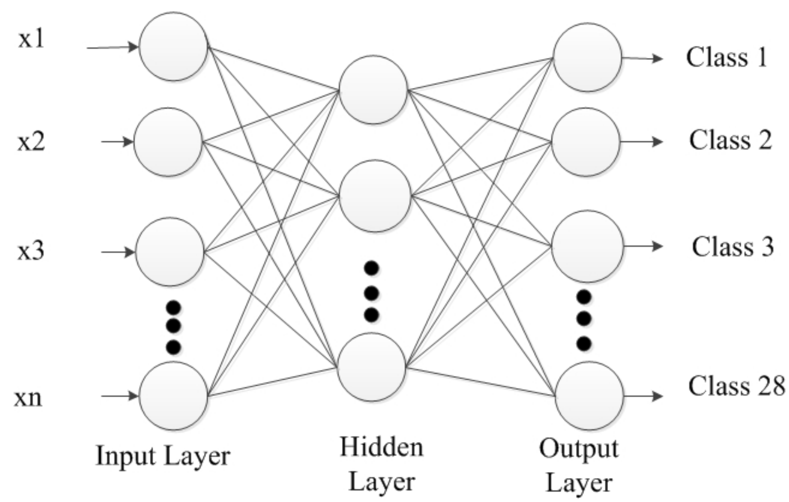


Figure 3.11: Basic Structure of multilayer neural network

Each connection to a neuron has an adjustable weight factor associated with it. Every neuron in the network sums its weighted inputs to produce an internal activity level given as:

$$a_i = \sum_{j=1}^{n} w_{ij} x_{ij} - w_{io} \tag{3.1}$$

where w_{ij} is the weight of the connection from input j to neuron i, x_{ij} is the input signal number j to neuron i and w_{io} is the threshold associated with unit i. The internal activity is passed through a nonlinear function φ to produce the output of the neuron y_i, where:

$$y_i = \varphi(a_i) \tag{3.2}$$

The weights of the connections are adjusted during the training process to achieve the desired input/output relation of the network. A multilayer feed forward network has its neurons organized into layers with no feedback or lateral connections. Layers of neurons other than the output layer are called hidden layers. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The back propagation algorithm [63] is a supervised iterative training method for multilayer feed forward nets with sigmoidal nonlinear threshold units. It uses training data consisting of input-output pairs of vectors that characterizes the problem. Using a generalized Least-Mean-Square algorithm, the back propagation algorithm minimizes the mean square difference between the real network output and the desired output [65]. The error function that the back propagation algorithm minimizes is the average of the square difference between the output of each neuron in the output layer and the desired output. The error function can be expressed as:

$$E = \frac{1}{P}\sum_{p}\sum_{k}(d_{pk} - o_{pk})^2 \tag{3.3}$$

where p is the index of the P training pair of vectors, k is the index of elements in the output vector, d_{pk} is the k^{th} element of the p^{th} desired pattern vector, and o_{pk} is the k^{th} element of the output vector when pattern p is presented as input to the network [66]. Minimizing the cost function represented by equation (3.3) results in an updating rule to adjust the weights of the connections between the neurons. The weight adjustment of the connection between neuron i in layer m and neuron j in layer $m+1$ can be expressed as:

$$\Delta w_{ji} = \eta \delta_j o_i,\ i,\ j = 1, 2, ..., N, \tag{3.4}$$

where N is the total number of units, i is the index of units in layer m, η is the learning rate, o_i is the output of unit i in the m^{th} layer and δ is the change in error term which is back propagated from the j^{th} unit in layer $m+1$ defined by:

$$\begin{aligned} \delta_j &= [d_j - o_j]o_j[1 - o_j] \\ &= y_j[1 - y_j] \sum_k \delta_k w_{kj} \end{aligned} \tag{3.5}$$

Neuron j is in a hidden layer and k is index of neurons in the layer $(m+2)$, ahead of the layer of neuron j. The MLP discussed is a typical one with enhanced version published throughout the last two decades. The basic MLP network has a number of attractive features but its two main ones are its ability for generalization even in the presence of high noise power on the observations. Moreover, it is fault tolerant; as the network keeps providing a good performance even when a significant fraction of its neuron and interconnection fail. It is worth noting, however, that The network has a number of limitations including high computational load, the problem of local minima, and scaling issues (i.e. from small to large scale systems).

Naive Bayes Classifier

To access the performance of the LMC and the extracted features across different classifiers, we also implemented the Naive Bayesian (NB) classifier. The NB classifier is a simple probabilistic classifier based on the famous Baye's theorem with the naive assumption of independence between every pair of features. The Bayesian classification approach is based on quantifying the trade-offs between various classification decisions using probabilities and the costs that accompany such decisions [66]. There are different types of Naive Bayes Classifier: The Gaussian, multinomial and the Bernoulli NB, among others. In multinomial NB, feature vectors represent frequencies in which certain events were generated while in Bernoulli NB, they are boolean in nature. These two cases does not apply to our collected data, where features are continuous values associated with each class. In addition, the histogram obtained for the different features can be approximated as Gaussian distribution. Some of these histograms are shown in Figures 3.12 and 3.13. Similar plots were also obtained for other features. Our particular setup consist of N features and K classes with N = 12 and K = 28. Given a class variable y_j (or just a given sign) and a given feature vector $x_1, ..., x_N$ (12 features), the Baye's theorem states that:

$$P(y_j|x_1, ..., x_N) = \frac{P(x_1, ...x_N|y_j)P(y_j)}{P(x_1, ...x_N)} \tag{3.6}$$

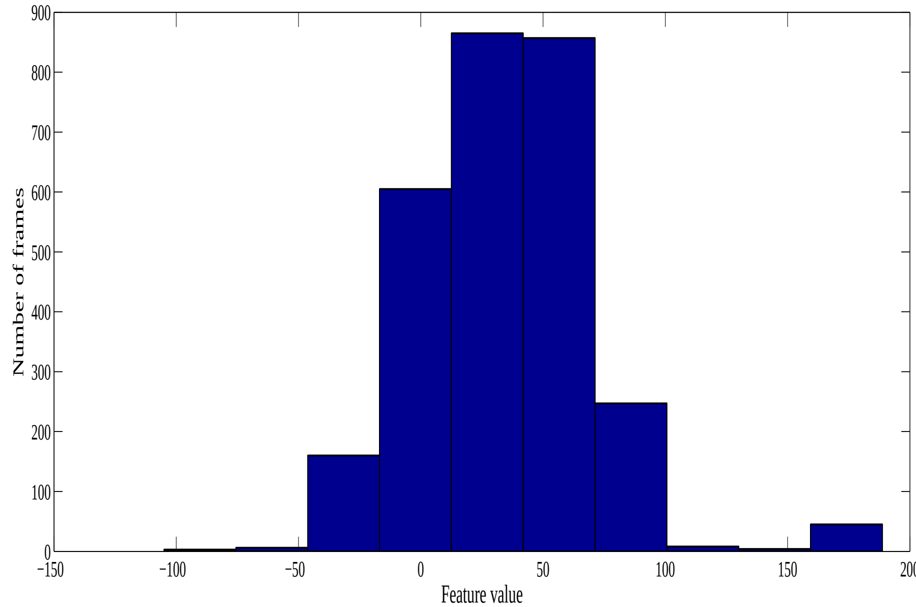


Figure 3.12: Histogram of feature 'hand palm position along x-axis'

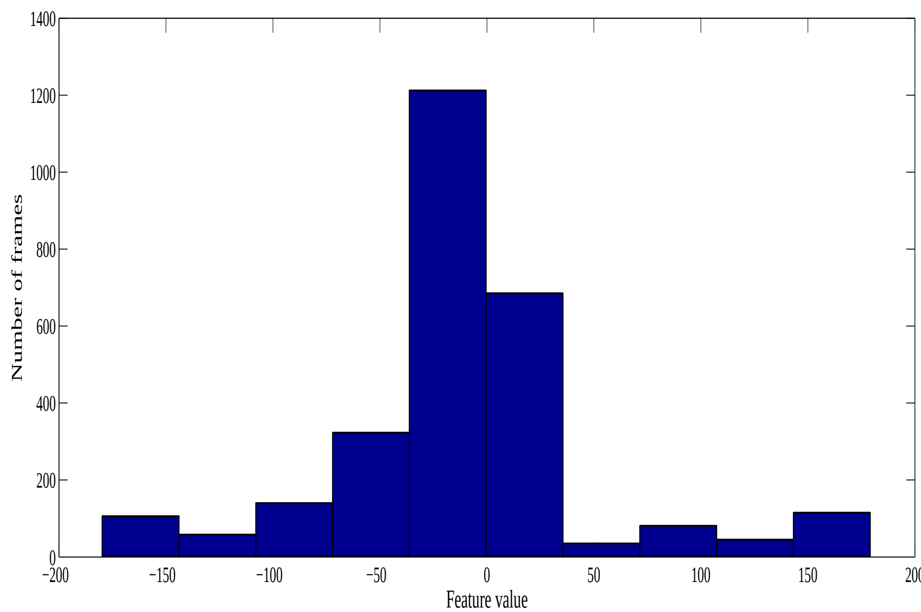


Figure 3.13: Histogram of feature 'hand roll'

where $j = 1, 2, ..., K$. Using the Naive independence assumption among the features, we can write:

$$P(x_i|y_j, x_1, ..., x_N) = P(x_i|y_j) \tag{3.7}$$

where $i = 1, ..., N$, and $j = 1, ..., K$ Hence,

$$P(y_j|x_1...x_N) = \frac{P(y_j)\prod_{i=1}^{N} P(x_i|y_j)}{P(x_1, ..., x_N)} \tag{3.8}$$

since $P(x_1, ..., x_N)$ is constant for a given input, we get:

$$P(y_j|x_1...x_N) \propto AP(y_j)\prod_{i=1}^{N} P(x_i|y_j) \tag{3.9}$$

where A is a scaling factor. From equation (3.9), for a given feature vector, $\mathbf{x} = [x_1, ..., x_N]^T$, the NB decision rule simply assigns $\mathbf{x}$ to class y_k if:

$$P(y_k)\prod_{i=1}^{N} P(x_i|y_k) > P(y_j)\prod_{i=1}^{N} P(x_i|y_j) \tag{3.10}$$

where $j = 1, ...k$, and $k \neq j$. If the classes have equal a priori probabilities, the rule is simplified to:

$$\prod_{i=1}^{N}(P(x_i|y_k) > \prod_{i=1}^{N} P(x_i|y_j) \tag{3.11}$$

where $j = 1, ...K$, and $k \neq j$. When the independence criteria does not apply, then we have the more general Bayesian classifier which uses the joint Gaussian distribution to model the statistical behavior of the feature vector. Similar to the MLP, the NB classifier exhibits a number of advantages and limitations. In particular, the NB classifier is fast to train and to classify. The algorithm is insensitive to irrelevant features. It can handle real, discrete data, and streaming data well. Its major disadvantage is the independence assumption, however, the NBC can be very robust to violations of its independence assumption, and it has been reported to perform well for many real world data sets [67]. In addition, it requires large amount of data to properly model the distribution of the different features under different classes.

The results obtained using one LMC setup are presented in Chapter 4. A major challenge we are faced with is finger occlusion within the LMC field of view. This affected substantially, the overall classification accuracy. As such, we started investigating the use of a pair of LMCs to improve the robustness of the whole system.

3.2 Alphabets Level Recognition using Two LMCs

In this section, the concept of using two LMCs is presented. This was proposed to give a combined view of the performed sign. The idea here is to place one LMC in front of the signer and the other at the side of the signer. The setup is as shown in Figure 3.14. Similar data collection process as presented in section 3.1 was carried out. However, in this case, we have used Linear Discriminant Analysis

Figure 3.14: Experimental set up using two LMCs

(LDA) as a preprocessing stage to reduce redundancy in the feature domain. We also used the LDA decompositionnin the classification stage with a basic euclidean distance. We then compared the combination of feature sets from the two LMCs and fusion of decisions at classifier level with that of evidence based fusion, namely the Dempster-Shafer (D-S) theory of evidence. Figures 3.15 and 3.16 show the two fusion approaches. This is followed by explanation of the LDA classifier and Dempster-Shafer theory. As compared to other classifier fusion algorithms used in the literature, D-S theory was chosen due to its wide spread application in modeling uncertainty [37, 68].

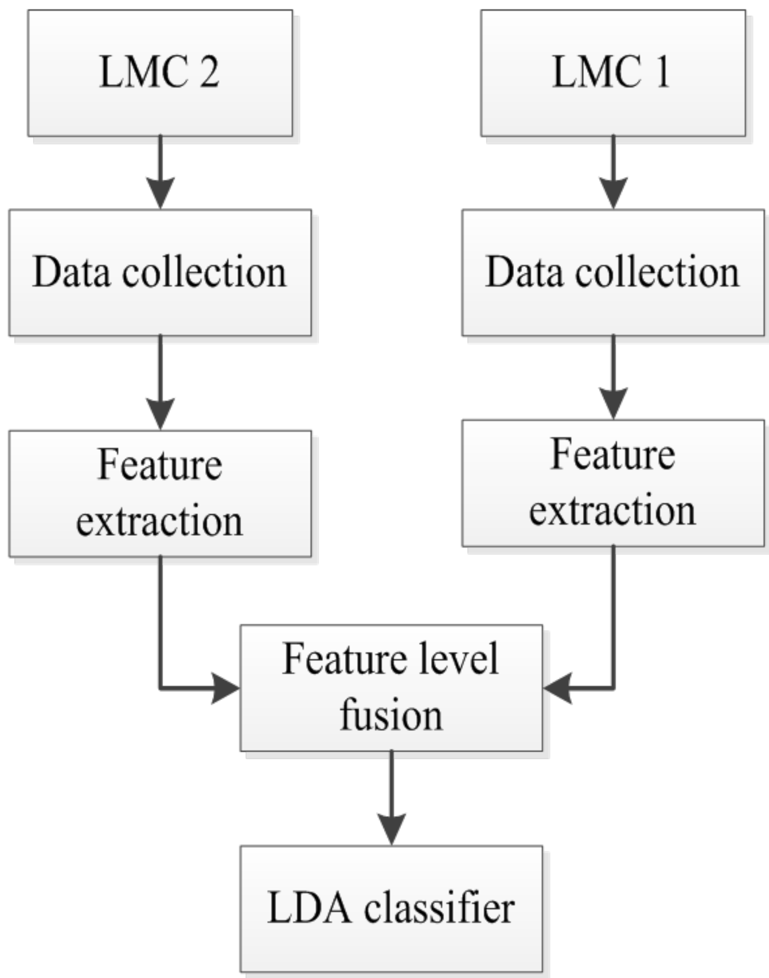


Figure 3.15: Feature level fusion

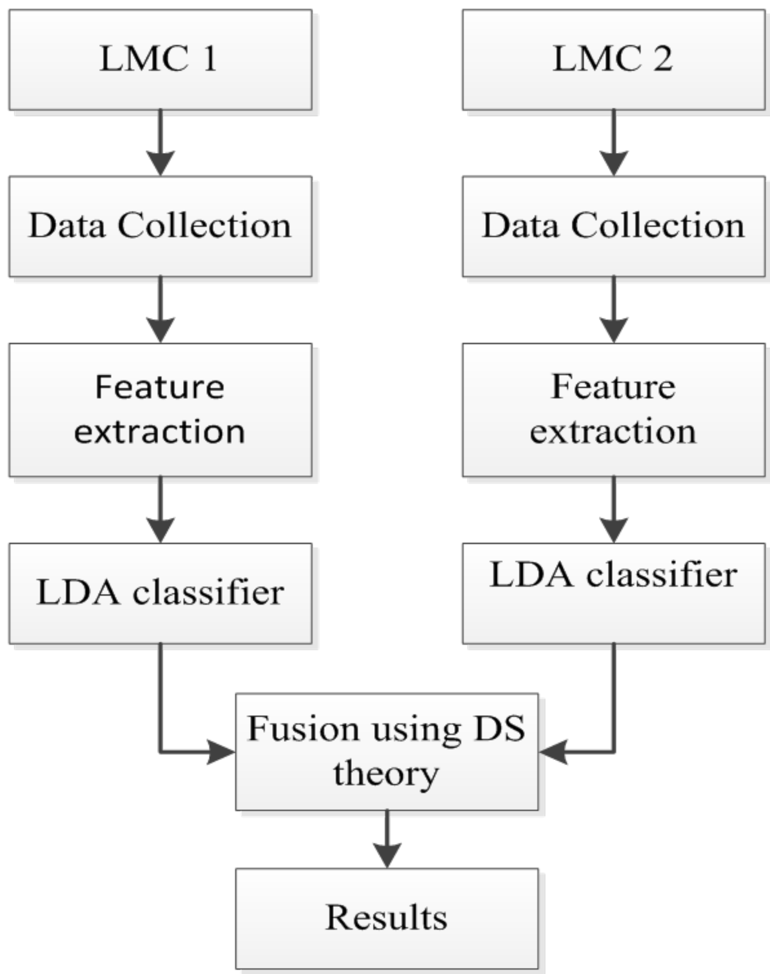


Figure 3.16: Decision level Fusion

3.2.1 Linear Discriminant Analysis

Linear Discriminant analysis is a very common statistical approach used in signal processing, PR and other machine learning applications [69]. The basic principle of LDA relies on projecting high dimension data onto a minimal possible dimension space such that the data can achieve maximum separation among different classes in the projected domain. Within-class scatter matrix, and between-class scatter matrix are two popular measures used to describe separability of classes in a given data set, where within-class scatter matrix is given by:

$$S_w = \sum_{j=1}^{M} \sum_{i=1}^{N} (\mathbf{x}_i^j - \mu_j)(x_i^j - \mu_j)^T \tag{3.12}$$

where $\mathbf{x}_i^j$ is the i^{th} sample vector of class j, $\boldsymbol{\mu_j}$ is the mean vector of class j, M is the total number of classes (i.e. 28 ArSL alphabets), and N is the total number of samples in class j. Similarly, between-class scatter matrix is given by:

$$S_b = \sum_{j=1}^{M} (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T \tag{3.13}$$

where $\boldsymbol{\mu}$ represents vector-mean across all classes and $\boldsymbol{\mu_j}$ is the mean vector for class j. The essence of using these measures is in finding a linear transformation which can maximize the inter-class variance, at the same time minimizing the intra-class variance.

3.2.2 Dempster-Shafer Theory of Evidence

The Dempster-Shafer (D-S) theory of evidence has been successfully introduced as a robust approach for fusing information from different experts. In our case, we use it to combine decisions from two different LMCs. The D-S theory is based on the concept of beliefs [58]. The theory was first introduced by Shafer and Dempster as a way to generalize Bayesian probability theory. It is commonly known and referred to as the theory of evidence for belief functions. Recall that in the case of Bayesian theory, the following equation must hold:

$$P(\mathbf{x}|C_1) + P(\mathbf{x}|C_2) + ... + P(\mathbf{x}|C_n) = 1 \tag{3.14}$$

where $\mathbf{x}$ is a vector and the C_n's represent a set of classes. The generalization of equation (3.14) obtained using the D-S concept is written as:

$$P(\mathbf{x}|C_1) + P(\mathbf{x}|C_2) + ... + P(\mathbf{x}|C_n) + P(\theta) = 1 \tag{3.15}$$

where $P(\theta)$ represents the uncertainty; hence, this technique is commonly used to model uncertainty. The theory is based on three basic concepts: basic belief assignment, a belief function and plausibility. The basic belief assignment (bba) serve as the basis of evidence theory from which belief function and plausibility are computed. It allocates a value which lies between 0 and 1 to each variable in subset (A) such that the bba value of the null set is 0 and the sum of the bba values of all subsets is sum up to 1. Evidence is considered to be certain if

$m(.) = 1$. In relation to this work, the posterior probabilities of classes obtained from the two LMCs classifiers serve as the bba (i.e. $m(.)$) from which plausibility and belief function are computed. The bba must satisfies the following conditions:

$$\begin{aligned} 0 \leq m(A) \leq 1 \\ m(\emptyset) = 0 \\ \sum_{A \in P(X)} m(A) = 1 \end{aligned} \tag{3.16}$$

where $P(X)$ is the power set of (X) and (A) is an element in the power set of (X). Meanwhile, the belief function assigns a value $[0, 1]$ to every non-empty subset (B). Two interval bounds can be defined for every probability assignment. The D-S theory represents the lower bound by belief function, which is evaluated as the sum of all of the bba of the proper subsets of (B). The top limit of the probability assignment is defined as plausibility. It is evaluated as the sum of all posterior probabilities of the sets (B) which intersect with the set of interest in set (A) [58].

$$Bel(A) = \sum_{B \subset A} m(B) \tag{3.17}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \tag{3.18}$$

where Bel represents the belief function and Pl represents the plausibility function. The combination rule expressed in equation (3.19) is used to combine all the evidence.

$$m_n(A) = \frac{\sum\limits_{A_1 \cap A_2 \cap ... \cap A_n} m_1(A_1)m_2(A_2)...m_n(A_n)}{1 - \sum\limits_{A_1 \cap A_2 \cap ... \cap A_n = \emptyset} m_1(A_1)m_2(A_2)...m_n(A_n)} \tag{3.19}$$

Equation (3.19) represents the combination rule for n-pieces of evidence. In the case of this work, n is two (i.e. decisions from the two LDA based classifiers). For more details on the D-S theory of evidence, the reader is directed to

3.3 Recognition of Isolated Word Signs

In this section, the two-LMC set up presented in section 3.2 was used for the recognition of 100 ArSL isolated words. The 100 words were drawn from the Unified Arabic Sign Language Dictionary in the categories: family, colors, food, religion, jobs, and title, etc. In what follows, we discuss the data collection stage, the feature extraction stage and finally the classification stage.

3.3.1 Data Collection for Isolated Sign Words

We have used a native deaf signer to perform the required signs. Ten observations of each sign were collected for 100 signs giving a total of 1000 observations. Each observation contains several frames of data representing the detected hand orientation. The number of frames in each observation depends largely on the length of the sign word. In addition, the ability of the LMC device to detect the performed sign also affects the numbers of frames that can be collected per observation. It was observed that the device can have difficulties detecting some of the performed signs, especially signs that are performed close to the face or the head.

To solve some of the problems with difficult signs, the signer was required to perform the sign repeatedly for a number of times ranging from two to five times. Out of the 100 performed signs, LMC 1, which is the LMC in front of the signer, detected 55 signs with 10 observations, while LMC 2, which is the LMC at the right side of the signer, detected 79 signs with 10 observations each. In summary, there were some signs which LMC 1 could detect while LMC 2 could not, and vice versa. While there were several signs in which both LMCs detect easily, while few cases where both LMCs fail to detect the signs. This was the motive behind the idea of using multiple LMCs, since a single LMCs cannot do well in detecting all signs. LMC 2 does well in detecting signs performed close to the signer face, while LMC 1 is good in detecting signs that requires the signer to stretch out his hand while performing the sign. Table 3.2 shows list of some of the ArSL words performed by the native deaf signer.

Table 3.2: List of ArSL words

S/N	List of signs	قائمة لغة الإشارة العربية
1	Family	عائلة
2	Green peas	بازلاء
3	Twins	توأمان
4	Son	ابن
5	Engagement	خطوبة
6	Death	وفاة
7	Color	لون
8	Five	خمسة

9	Golden	ذهبي
10	Jeweler	صائغ
11	Interpreter	مترجم
12	Sign language interpreter	مترجم لغة الإشارة
13	Teacher	معلم
14	Chief of parliament	رئيس البرلمان
15	Blacksmith	حداد
16	dyer	صباغ
17	Tailor	خياط
18	Pilot	طيار
19	Cook	طبخ
20	Banana	موز
21	Watermelon	بطيخ أحمر
22	Cauliflower	قرنبيط
23	Zucchini	كوسة
24	Garlic	ثوم
25	Bread	خبز
26	Milk	حليب
27	Breakfast	الإفطار
28	Muslim	مسلم
29	Allah	الله
30	Messenger	رسول
31	Prophet Isah (AS)	النبي عيسى
32	Chapter of Quran	سورة من القرآن
33	Verse of Quran	آية من القرآن
34	Hajj	الحج
35	Lesser hajj	العمرة
36	Ablution	وضوء
37	Dry ablution	التيم
38	Rubbing sock	مسح علي الخفين
39	Nullification of prayer	مبطلات الصلاة
40	Leader	إمام

41	Orator	خطيب
42	New year	عيد رأس السنة
43	Church	كنيسة
44	Zero	صفر
45	Four	أربعة
46	Ten	عشرة
47	Thirty	ثلاثون
48	Forty	أربعون
49	Fifty	خمسون
50	One hundred	مائة
51	Three hundred	ثلاثمائة
52	Five hundred	خمسمائة
53	Six hundred	ستمائة
54	Seven hundred	سبع مئة
55	Eight hundred	ثمانمائة
56	Nine hundred	تسعمائة
57	Ten Thousand	عشرة آلاف
58	One hundred thousand	مئة الف
59	One million	مليون
60	One billion	مليار
61	Snake	ثعبان
62	Cable	كعكة
63	Bear	دب
64	Crocodile	تمساح
65	Shark	سمك القرش
66	Whale	حوت
67	Elephant	فيل
68	Gorilla	غوريلا
69	Giraffe	زرافة
70	Falcon	صقر
71	Eagle	نسر
72	Cock	الديك

73	Bee	نحلة
74	Wall	جدار
75	Room	غرفة
76	Bedroom	حجرة النوم
77	Bed	سرير
78	Bed spread	انتشار السرير
79	Kitchen	مطبخ
80	Gas stove	فرن غاز
81	Plate	طبق
82	Glass cup	فنجان
83	Freezer	مجمد فريزر
84	Dinning room	غرفة الطعام
85	Plug	قابس كهرباء
86	Table	طاولة
87	Chair	كرسي
88	Carpet	سجادة
89	Chandelier	ثريا
90	Television	تلفزيون
91	Video camera	كاميرا فيديو
92	Photo camera	كاميرا فوتوغرافيه
93	Long	طويل
94	Guest room	غرفة الضيوف
95	Fan	مبخرة
96	Heater	مدفأة
97	Key	مفتاح
98	Air conditioner	مكيف هواء
99	Electricity	كهرباء
100	Friday Sermon	خطبة الجمعة

3.3.2 Feature Extraction

From the collected data, we collect over 24 attributes describing the hands orientation, these include the 12 features used in the alphabet level recognition. However, not all are relevant to perform sign classification because the total number of collected attributes varies with respect to the number of hands detected or involve in performing the signs. It should be noted that some signs involve using two hands while some signs involve a single hand. If the sign involves one hand, the right hand is used. Similar analysis as previously discussed was carried out to select the most discriminative features, from which we selected 16 as our feature vector for sign classification. List of features are shown in Table 3.3.

Table 3.3: List of Features

S/N	Feature Name
1	Translation probability
2	Finger length
3	Finger width
4	Average tip position along x-axis
5	Average tip position along y-axis
6	Average tip position along z-axis
7	Hand sphere radius
8	First hand palm position along x-axis
9	First hand palm position along y-axis
10	First hand palm position along z-axis
11	Second hand palm position along x-axis
12	Second hand palm position along y-axis
13	Second hand palm position along z-axis
14	Hand pitch
15	Hand roll
16	Hand yaw, etc.

The histograms of some of these features are shown in Figures 3.17, 3.18 for LMC 1.

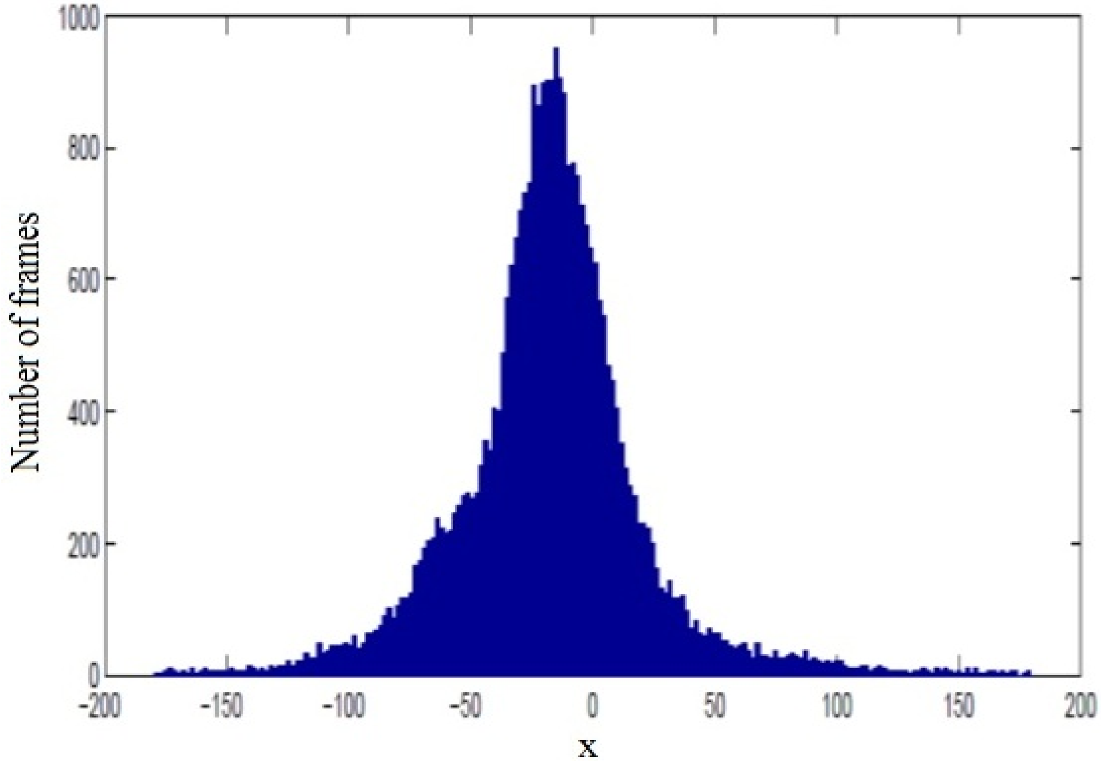


Figure 3.17: Histogram of hand pitch from LMC 1, for the 100 signs

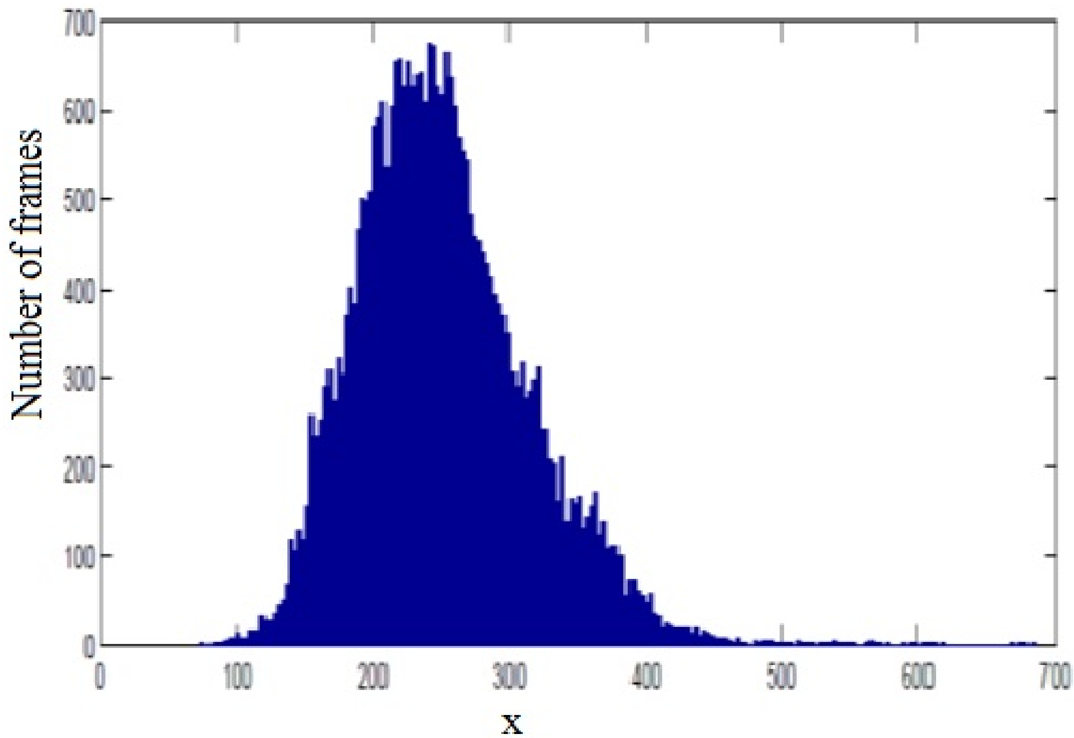


Figure 3.18: Histogram of hand palm position along y-axis in LMC 1 for the 100 signs

Similar histograms were obtained for the case of LMC 2 as shown in Figures 3.19 and 3.20.

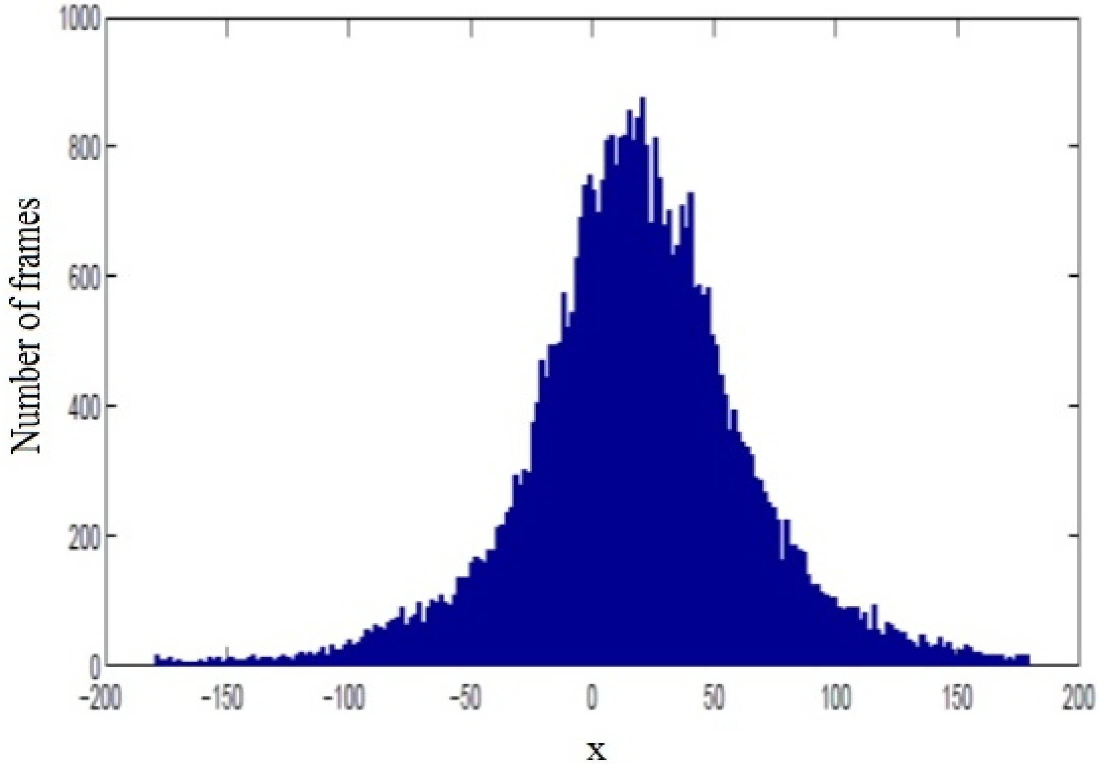


Figure 3.19: Histogram of hand pitch from LMC 2, for the 100 signs

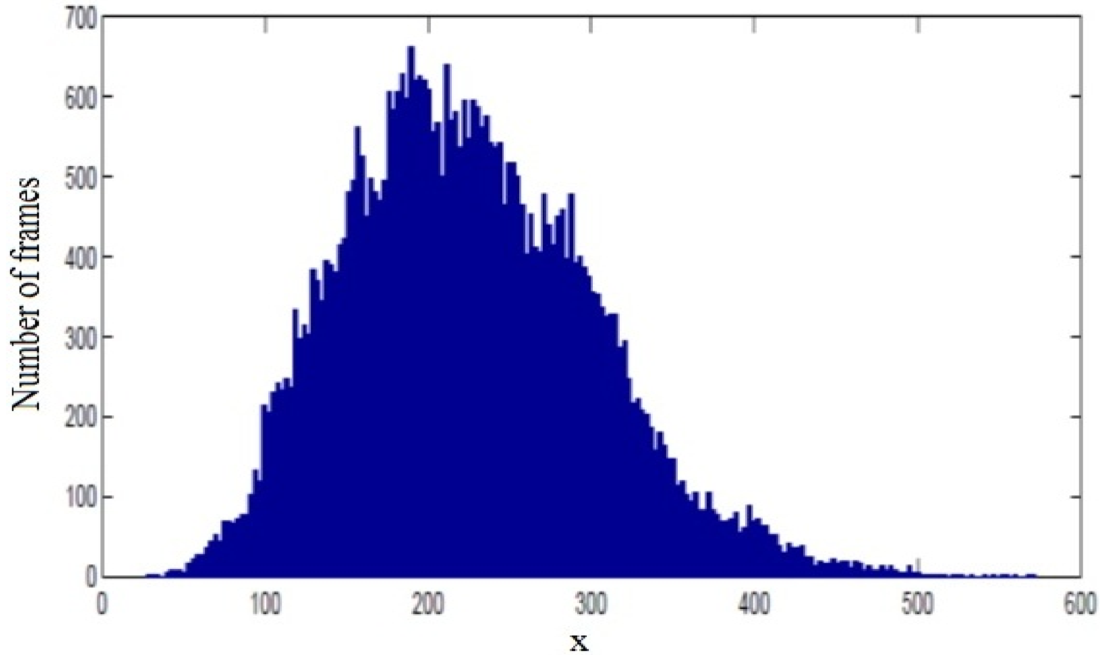


Figure 3.20: Histogram of hand palm position along y-axis in LMC 2 for the 100 signs

From the observed histograms, we opted to use a Gaussian Mixture Model (GMM) for representing the features statistical behaviour. Hence, a Baye's classifier was used with the training phase based on the GMM.

3.3.3 Classification of Isolated Word Signs

Unlike the case of alphabet sign recognition where all consecutive frames obtained for a particular sign represents the sign, frames obtained in the case of isolated signs represent the different sequences involved for the performed sign. Consequently, this makes the task of classification more challenging and different from the case of alphabet recognition. Similar to previously discussed setup, we have considered the shape of the histogram of features obtained to decide the distribution type. From the histogram of features obtained, we used a GMM Baye's classifier for classification of the 100 signs. A GMM is a parametric probabilistic model represented by sum of weighted Gaussian component densities. GMMs have been widely used as robust parametric statistical models for diverse applications including biometric system, speech spectral features, etc [70].

The Gaussian Mixture Model

A Gaussian mixture consists of k different Gaussian distributions, with k being specified by the user. Let x denote a given feature. Following is a univariate Gaussian distribution:

$$g(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{(x-\mu)^2}{2\sigma^2}] \tag{3.20}$$

This distribution is generalized to the multiple variables case for a d-dimensional vector $\mathbf{x}$ as:

$$g(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp[\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})] \tag{3.21}$$

where $\boldsymbol{\mu}$ is the mean vector, and $\boldsymbol{\Sigma}$ is the $d \ X \ d$ covariance matrix. The GMM model uses multiple distributions of the function above. Given a particular dataset, the task is to estimate $(\boldsymbol{\Sigma}, \boldsymbol{\mu})$ and the weight (π_k) of each distribution. In the case of a single Gaussian, we can use the Maximum Likelihood (ML) principle to estimate the mean vector and the covariance matrix. We start by taking the *log* of the Gaussian distribution in equation (3.21) to get:

$$\ln P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(x - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \tag{3.22}$$

Taking the derivative with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and equating to zero we get:

$$\frac{\partial \ln P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}} = 0 \tag{3.23}$$

$$\frac{\partial \ln P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} = 0 \tag{3.24}$$

From equations (3.23) and (3.24), we can estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{3.25}$$

and

$$\boldsymbol{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})(x_n - \mu_{ML})^T \tag{3.26}$$

where N is the number of data points (or observations). In the case of multiple Gaussians mixture, we have:

$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k g(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.27}$$

where K is the total number of Gaussian mixtures and $g(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the normal multivariate Gaussian distribution for the k^{th} Gaussian, π_k is the weight for each of the mixture and satisfies the following conditions.

$$
\begin{aligned}
0 \leq \pi_k \leq 1 \\
\sum_{k=1}^{K} \pi_k = 1
\end{aligned}
\tag{3.28}
$$

Taking the log-likelihood of equation (3.27), we get:

$$
\begin{aligned}
\ln P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi) &= \sum_{n=1}^{N} \ln P(\mathbf{x}_n) \\
&= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k g(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)
\end{aligned}
\tag{3.29}
$$

N is the number of data point. In this case, ML does not work as there is no closed form solution for equation (3.29). Therefore, parameters can be estimated using Expectation Maximization (EM) technique.

Expectation Maximization Algorithm

For a given dataset, the number of Gaussian components for optimal performance is given by the user, while the missing weights can be thought of as prior probabilities for the different components. For a given feature vector $\mathbf{x}$, we can evaluate the corresponding posterior probabilities called responsibilities. From Baye's rule, we define:

$$
\begin{aligned}
\gamma_k(\mathbf{x}) &= P(k|\mathbf{x}) \\
&= \frac{P(k)P(\mathbf{x}|k)}{P(\mathbf{x})} \\
&= \frac{\pi_k g(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j g(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}
\end{aligned}
\tag{3.30}
$$

where γ_k is a latent variable and $\pi_k = \frac{N_k}{N}$, N_k is the effective number of points assigned to cluster k.

EM algorithm is an iterative optimization technique which is operated locally to find out the set of parameters in GMM distribution [71]. For K-Gaussian, we need to estimate K-weights, K-set of means, which have the dimension of the data, and K-set of covariance matrices. There are two steps involve in EM:

1. Estimation step:- for a given parameter values, we compute the expected values of the latent variables.

2. Maximization step:- updates the parameter of the model based on the obtained latent variable using ML method.

EM Algorithm for GMM

Given a Gaussian Mixture Model (GMM), the goal is to maximize the likelihood function with respect to the parameters comprising the means and covariance of the components and the mixing weights. The following steps are involve in finding the parameters.

1. Initialize the means, covariance and mixing coefficients, and evaluation of the log likelihood. Arbitrary set of values can be used as initial parameters.

2. E-step: estimate the responsibilities using the current parameter values as in equation (3.30).

3. M-step: Re-estimate the parameters using the current responsibilities (using equations (3.31), (3.32) and (3.33)).

4. Evaluate the log-likelihood using equation (3.29), if there is no convergence, return to step 2.

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)} \tag{3.31}$$

$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \boldsymbol{\mu}_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)} \tag{3.32}$$

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n) \tag{3.33}$$

Convergence is checked using some convergence criteria. For example, if the parameters do not change over a certain number of iteration or if the difference between the current estimated parameters and the previous is below a certain threshold, the algorithm can be considered to have converged. In our particular case, an error threshold of $1x10^{-6}$ was used with a maximum allowable number of iteration with which the error margin should be achieved. The algorithm is exited if the error margin is not achieved after the maximum number of iteration is reached.

GMM Bayes classifier

As discussed in section 3.3.1, each observation of a sign contains several frames of data depending on the length of the sign. The frames represent the sequences of the performed sign. Therefore, to recognize a sign, we need to compute the probability that all the sequences of a particular test sample belong to a particular sign. First, using the GMM algorithm previously discussed, M models were built using 70% of the data set, where M is the total number of signs available. The M models are then used for testing as follows:

For a given observation o, which consist of sequences of frames for the performed sign, the probability that the observation belongs to a particular sign is expressed as:

$$P(C_j|o) = \max_k(\log P(\mathbf{x}_1|M_k) \times P(\mathbf{x}_2|M_k) \times ... \times P(\mathbf{x}_n|M_k)) \tag{3.34}$$

Where C_js are the different signs, $k = 1, 2, 3...100$, n is the total number of frames in the observation, and $j = 1, 2, 3, .., 100$. $P(\mathbf{x}_n|M_k)$ is given as:

$$P(\mathbf{x}_n|M_k) = \sum_{j=1}^{J} \pi_j g(\mathbf{x}_n|\boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j) \tag{3.35}$$

J is the total number of Gaussian mixtures used and $g(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for each of the components is given by:

$$g(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp[\frac{-1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)], \tag{3.36}$$

where d is the feature dimension which is 16 in our case.

To select the 70% training data from the data set, we selected, randomly, 7 observations out of 10 from each of the signs. To avoid fitting the data to a particular set of testing observations, this was repeated for different numbers of runs, and different number of Gaussian mixtures. The results for the experiments are discussed in the next chapter.

3.4 Summary

In this chapter, we have presented three different experimental setups for recognition of ArSL: single LMC set up for alphabet level, double LMC setup for alphabet level and double LMC setup for recognition of isolated words. The different methods and algorithms used for analysis and classification were presented. In the next chapter, results obtained from the different setups will be presented and discussed.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, the results obtained for the various setups presented in chapter 3 are presented under the following three categories: recognition of 28 Arabic sign language alphabets using a single LMC, recognition of 28 Arabic sign alphabets using two LMCs and finally, recognition of isolated Arabic sign language words. These categories are presented herewith.

4.1 Results of Alphabets Level Recognition using One LMC

As previously discussed in chapter 3, in our experimental setup, we started by considering the Arabic alphabet signs. In this setup, the signs were performed using one signer. The signer was asked to repeat each sign letter 10 times, making a total of 280 observations. The training and testing of both the MLP and NBC classifiers were carried out using cross validation.

In particular, we used a 5 fold cross validation approach with final results averaged over 5 runs. After acquiring the data from the LMC using NetBeans IDE interface, the data consisting of all the features for different signs was exported to WEKA machine learning software for classification. The WEKA package is an open source data mining and machine learning software package implemented in JAVA at the University of Waikato [72]. It was designed to make easy a quick try-out, using existing methods on new datasets in a flexible ways [73].

With the cross validation procedure discussed above, we reached an overall recognition accuracy of 95.5% with the NB classifier while the MLP provided an accuracy of 94.25%. The results are summarized in Table 4.1 for NBC, and Table 4.2 MLP classifiers respectively.

Table 4.1: Classification results of NBC

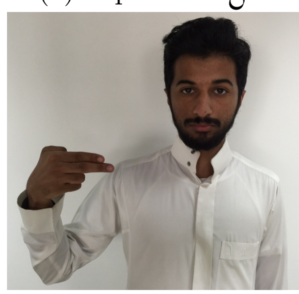
Correctly classified instances	2674, (95.5%)
Incorrectly classified instances	126, (4.5%)
Mean absolute error	0.003
Root mean squared error	0.05
Relative absolute error	4.81%
Root relative squared error	29.31%
Total number of instances	2800

Table 4.2: Classification results of MLP

Correctly classified instances	2639, (94.25%)
Incorrectly classified instances	161, (5.75%)
Mean absolute error	0.005
Root mean squared error	0.057
Relative absolute error	8.29%
Root relative squared error	30.45%
Total number of instances	2800

We also show in Tables 4.3 and 4.4, the letters that were misclassified the most. Actually, we can easily see from Figure 3.5 how the letter ض can be interpreted as ع (see Figure 4.1). Similar comments can be made for ح and خ as well as ث and ك.

(a) Alphabet "ض"



(b) Alphabet "ع"

Figure 4.1: Image showing typical misclassified alphabets

Table 4.3: Some of the misclassified letters from NBC

Actual letter	Misclassification error (%)	Misclassified as	At the rate (%)
ض	11	ع	100
ع	26	ض	76.9
ف	13	ع	100
ي	13	م	84.6
ص	8	ع	87.5

Table 4.4: Some of the misclassified letters from MLP

Actual letter	Misclassification error (%)	Misclassified as	At the rate (%)
ث	28	ت	25
ث	28	ك	32.1
ث	28	خ	35.7
ح	55	ج	32.7
ح	55	خ	36.4

The overall classification performance for both classifiers (NBC and MLP) is shown form the confusion matrices in Figures 4.2 and 4.3, respectively.

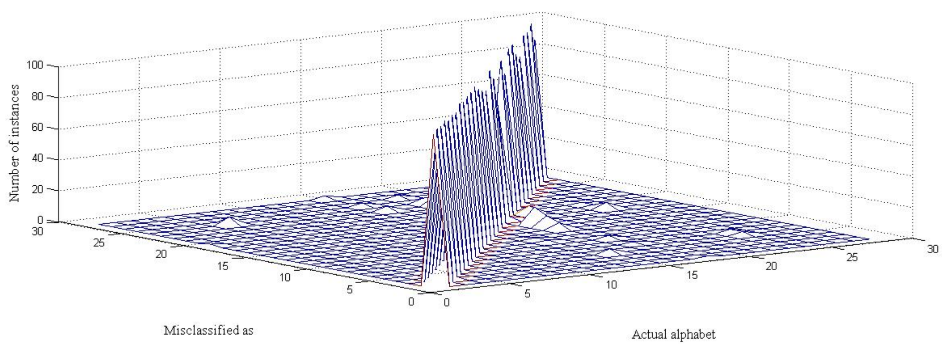


Figure 4.2: Confusion matrix for NBC classifier

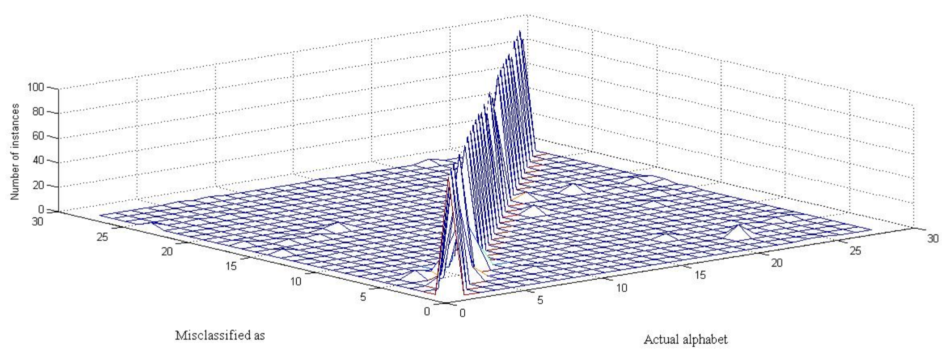


Figure 4.3: Confusion matrix for MLP classifier

While recognition accuracy has traditionally been used as a simple yet efficient measure of performance, it does not reveal the full performance of classification algorithms. Consider, for example, the case of 2 classes: object present and object absent (e.g, class A vs. class B). Assume that the database consists of 200 samples, of which 160 are B and 40 are class A. If we assume that all samples are classified as class B, then we have a classification accuracy of 80% which obviously does not make sense. For this reason, among others, researchers especially those working in medical area, tend to use a more comprehensive approach to describe accuracy, namely the Receiver Operating Characteristic (ROC).

In signal processing, the ROC is obtained by varying a given threshold and plotting the probability of detection (saying object present when it is in fact present) vs. the probability of false alarm (saying object present when in fact it is not). In practical diagnosis testing, the ROC basically plots the sensitivity (or True Positive Rate) of a given test as a function of the specificity (or False Positive Rate). For our experiment setup, we produced the ROC curve for each of the letters using a one-class versus all approach. The ROC curves for a number of letters using both the NB and the MLP classifiers are shown in Figures 4.4, 4.5, 4.6, 4.7 and 4.8, 4.9, 4.10, 4.11 respectively. The figures show that some letters 'ب', 'ع', and 'ف' produce excellent performance which is not the case for 'ح' and 'ث'. In the case of ب, the alphabet was 100% correctly classified, therefore, the Area Under Curve (AUC) of the ROC plot in Figure 4.4 is 1. For cases where the individual alphabets recognition accuracy is less than 100%, the ROC plots are not as smooth as the case of ب and hence, AUC less than 1.

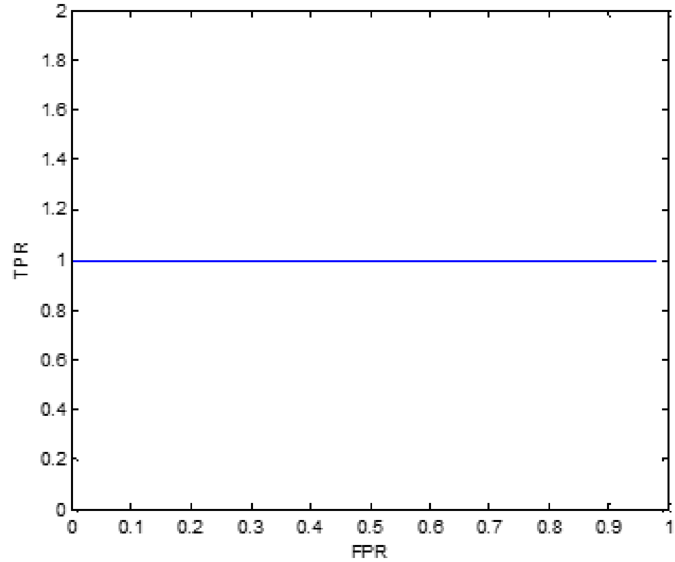


Figure 4.4: NBC classifier ROC for letter ب

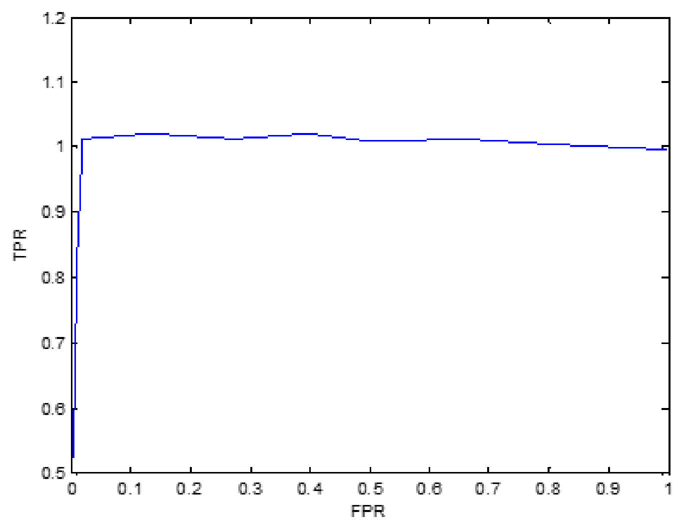


Figure 4.5: NBC classifier ROC for letter ع

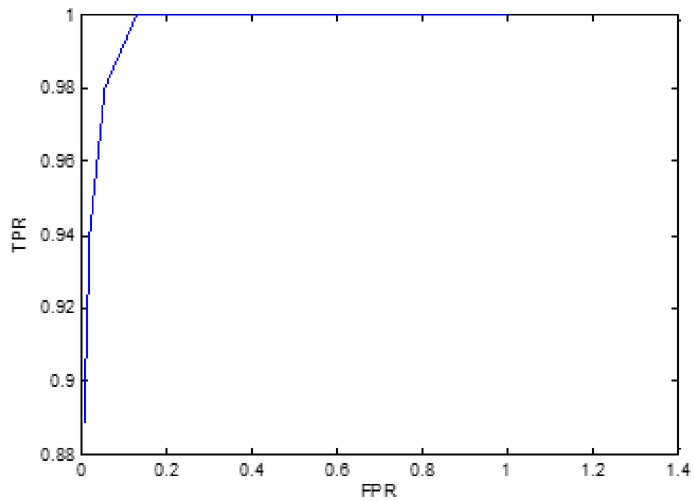


Figure 4.6: NBC classifier ROC for letter ف

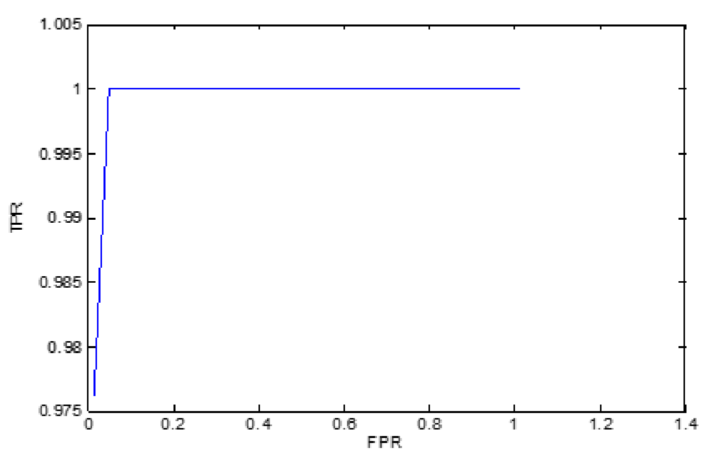


Figure 4.7: NBC classifier ROC for letter ل

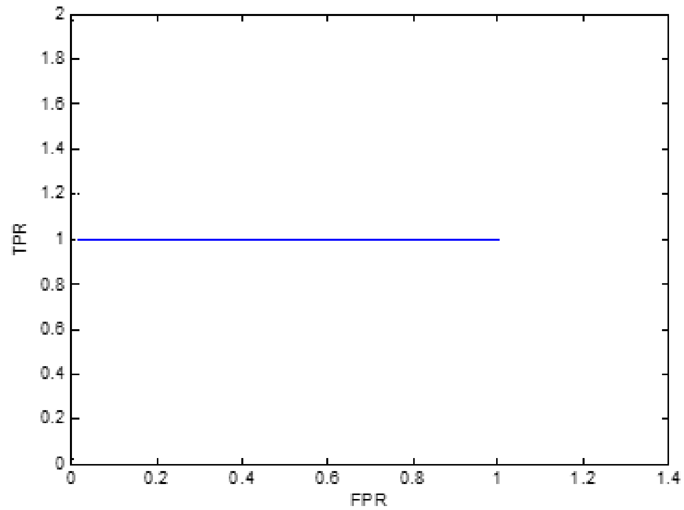


Figure 4.8: MLP classifier ROC for letter ب

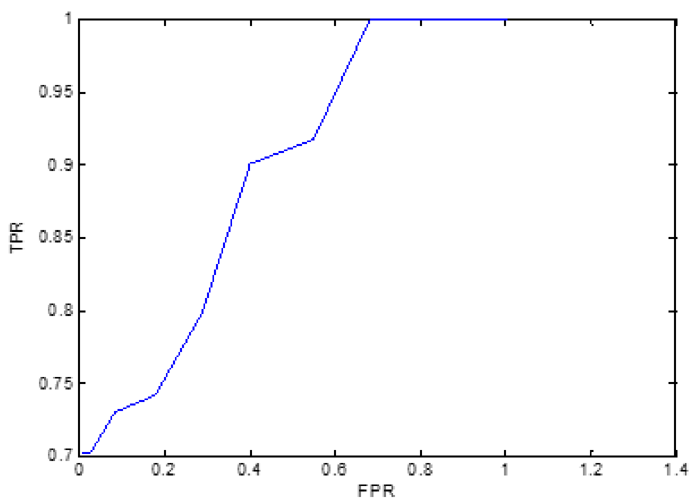


Figure 4.9: MLP classifier ROC for letter ث

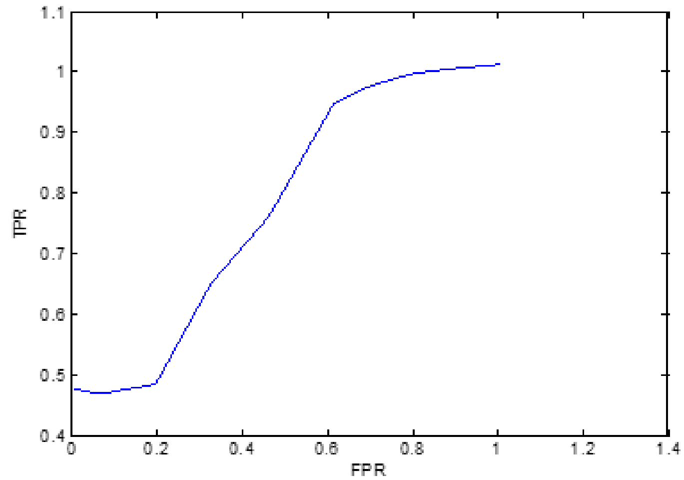


Figure 4.10: MLP classifier ROC for letter ح

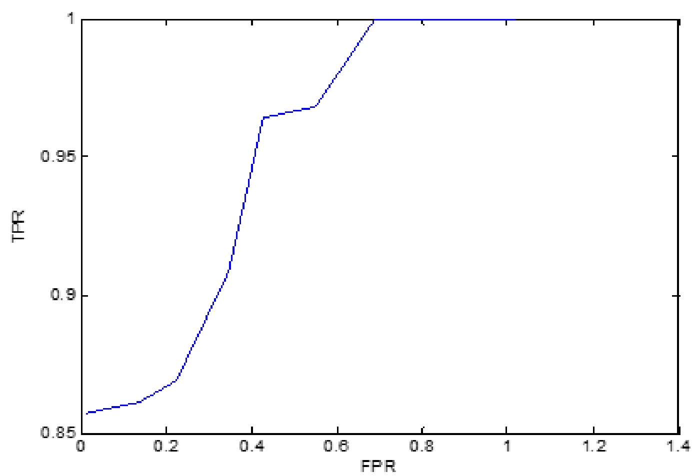
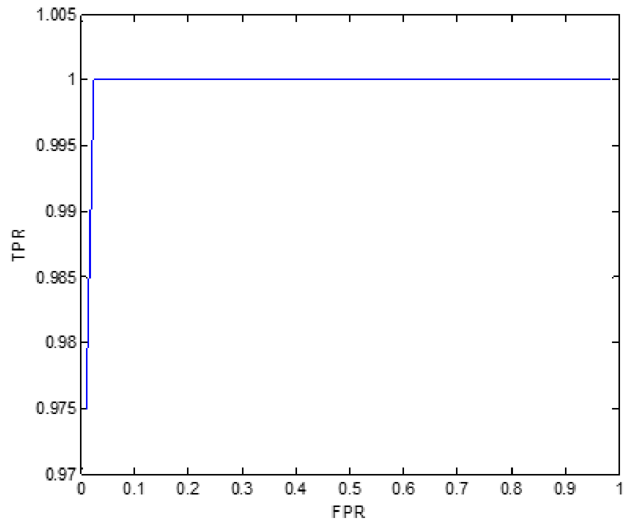
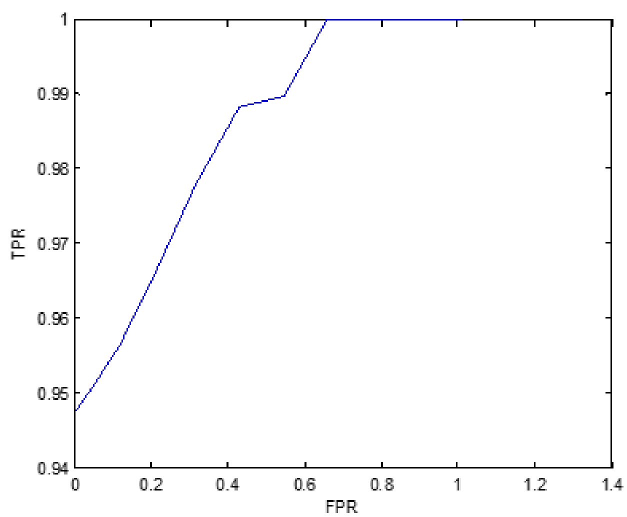


Figure 4.11: MLP classifier ROC for letter ض

Similar ROC plots were obtained for other alphabets. We have selected the ones shown here to show some cases of alphabets where the classifiers perform well and cases where they performed below 100%. We also show in Figure 4.12, the average ROC curve over the 28 letters. It is clear that NB classifier provides a better ROC curve than the MLP. Actually, the average ROC area or AUC of the NB classifier is 0.994 while it is 0.981 for the MLP classifier.



(a) Average ROC for NBC



(b) Average ROC for MLP

Figure 4.12: Average ROC curve of MLP and NBC classifier

The above results establish the proof of concept of using the LMC for ArSLR. However, in this case, we observed challenging cases where we have finger occlusion and hence, we further enhanced the system by considering two LMCs placed in a perpendicular position with respect to each other.

4.2 Results of Alphabets Level Recognition using Two LMCs

In this section, results obtained from combining two LMCs (one in front and the other at the side of the signer) are presented. We have used an LDA based classifier as well as D-S theory for fusion of decision from both LMCs. Any other classifier which can output probability can also be used. We compared fusion using D-S theory with feature level fusion (i.e. concatenating the two sets of 12 feature vector to form a 24 dimensional feature vector).

To test the algorithm, we started by splitting the available data (for all the alphabets) into 70% for training and the remaining 30% for testing. The test was repeated 10 times with the training and test set randomly selected on each run to avoid biasing to a particular test set. The results obtained from all the runs were averaged to obtain the final accuracy which are in three categories: classification results from individual LMCs, result from fusion of features of the two LMCs devices and results from fusion of classifiers using D-S theory. These three categories are summarized in Table 4.5.

Table 4.5: Summary of LDA classifier results

Runs	LMC 1 (%)	LMC 2 (%)	Feature combination (%)	D-S theory (%)
1	93.00	89.00	97.29	97.14
2	93.42	90.42	97.57	96.71
3	92.00	90.71	98.43	97.00
4	93.28	88.28	98.43	96.57
5	94.71	89.85	97.43	96.85
6	93.57	91.42	96.43	98.00
7	91.80	90.40	97.86	96.28
8	93.00	89.71	98.29	98.28
9	92.14	89.71	97.43	97.42
10	93.85	89.57	97.71	96.28
Average	93.08	89.91	97.69	97.05

From the 10 runs performed, the front LMC gave an average of 93.08% accuracy, while the side LMC gave 89.91%. Combination of features from the two LMCs gave an average of 97.69% accuracy while classifier level fusion using D-S theory gave 97.05%. As can be seen from Table 4.5, classifier and feature level fusion of evidence gave an improved recognition performance of the ArSLR system as compared to the individual LMCs. Fusion at feature level misclassified 44 instances while classifier level fusion misclassified 80 instances out of 2800 total instances. Some of the misclassified letters are shown in Tables 4.6 and 4.7.

Table 4.6: Some of the misclassified letters from feature fusion

Actual letter	Misclassified as	Number of times
أ	ع	8
ج	ق	8
ح	خ	4
ع	ل	4
ص	ش	4

Table 4.7: Some of the misclassified letters from classifier fusion

Actual letter	Misclassified as	Number of times
أ	ع	4
ظ	ب	8
ش	ص	12
ج	م	8
ب	ق	4

By analyzing the misclassified signs, it was observed that most of the misclassified signs are similar to the signs they are classified to. In addition, the results show an improvement over using a single LMC unit. From results obtained so far, it can be concluded that using two LMCs has advantage over one LMC. Therefore, we proceed to use the two LMC model for recognition of isolated Arabic sign language words.

4.3 Results of Isolated Word Sign Recognition using Two LMCs

The case of isolated signs recognition is more challenging than static signs (alphabet level) where both LMC respond to all signs and all consecutive frames represent the sign in the scene. Therefore, we propose a new recognition model for this case. In what follows, we present the model used for classification, followed by results obtained using the model.

4.3.1 Recognition Model

As discussed in chapter 3, we have used GMM for the case of dynamic sign classification. To perform sign classification, we developed a recognition system based on four classifiers model as shown in Figure 4.13. The model's decisions are sum up to form the final decision. The system works according to how well the LMCs device can detect the performed signs. The decision on which model to use was done by using three different thresholds, according to Figure 4.13. If LMC 1 has the best performance in detecting the performed sign, we rely on LMC 1 classifier for the final decision, likewise, if LMC 2 has the best performance in detecting the performed sign, final decision is based on LMC 2. However, if both LMC performed well on the sign, we combine evidences from the two LMCs to make the final decision.

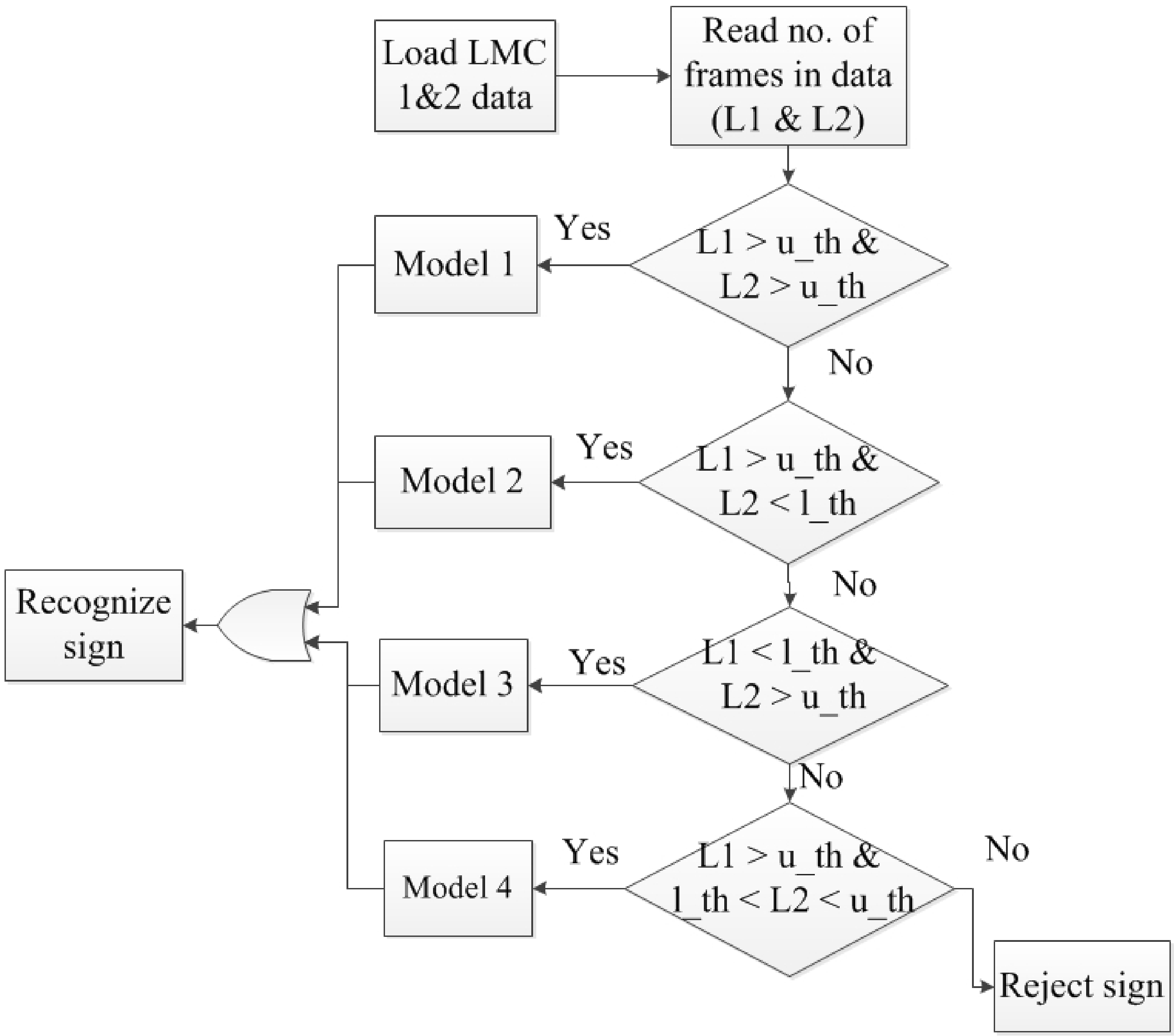


Figure 4.13: Model flowchart

From Figure 4.13, we have four classifier models, which are combined to form the overall system. The models are selected based on the following parameters: L_1, L_2, u_{th}, l_{th}, fr_{diff}. L_1 is the total number of frames in each observation of LMC 1 data, likewise L_2 for LMC 2. u_{th} is upper limit threshold, while l_{th} is lower limit threshold. fr_{diff} is used for the case where LMC 1 & 2 both detect the performed sign, to know how much the number of frames in LMC 1 is more than LMC 2, or vice versa. The following explain each of the model.

- Model 1: In this model, both LMC have sufficient data to make decision, therefore, we combine decision from both LMC 1 and 2. To solve the issue of different length of signs, we normalize signs to have equal length by down sampling. Other condition that fall into this category is when $u_{th} > L_1 > l_{th}$ & $u_{th} > L_2 > l_{th}$.

- Model 2: Here, LMC 1 was able to track the sign, while LMC 2 did not detect the sign or detect it with very few frames. Hence, we give decision confidence to LMC 1 only. Other option included in this model is the case where $u_{th} > L_1 > l_{th}$ & $L_2 < l_{th}$.

- Model 3: This model is the reverse case of model 2. LMC 2 is good in detecting the sign while LMC 1 is not. Hence we give confidence to LMC 2. Similarly, we have this condition $u_{th} > L_2 > l_{th}$ & $L_1 < l_{th}$ in this category. finally we have:

- Model 4: In this model, both LMCs tracked the performed signs with close range of frames. Therefore, we implemented a weighted combination of LMC

1 and 2. More weight is given to the LMC that has the highest number of frames. We also have this condition $L_2 > u_{th}$ & $u_{th} > L_1 > l_{th}$ under this model.

Apart from these four cases, the sign is rejected, (i.e. system cannot decide).

The algorithm was motivated by the fact that different signs have different time to perform the sign, hence, creating the possibility that some signs may have more numbers of frames than the other. In addition, it was also observed that the LMC respond to some signs with varying number of frames depending on how easy the device could track the performed sign. This has to do with the tracking capability of the LMC.

The proposed model in Figure 4.13 is not particular for the data set used in this work. It can be used for any data set collected using two LMCs setup. In future work, it will be used for other data set for the case of signer independent recognition.

4.3.2 Results of Isolated Sign Words

The proposed model was applied on a data set of 100 signs, with 10 observations per sign, obtained from two LMCs. The signs were performed by a native deaf signer. Results are presented for different number of Gaussian mixtures, and thresholds. Tables 4.8, 4.9, and 4.10 present results for case where $u_{th} = 10$, $l_{th} = 5$ and $fr_{diff} = 30$. We started by using $u_{th} = 10$ because on average, each of the observation contains 10 frames. Some signs have more than 10 frames per observation while some have less than 10. This is primarily due to variation in time length taken to perform signs. Therefore, we will compare accuracies as we go above $u_{th} = 10$ and below it. In this case, LMC 1 detects 21 signs out of

Table 4.8: Classification accuracy using 3 mixtures

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	14.00	30.00	69.00	63.67
2	15.00	32.00	69.67	64.33
3	17.67	32.00	70.67	65.33
4	15.00	31.33	67.67	64.67
5	15.67	32.33	70.33	66.33
6	18.00	31.67	71.33	66.67
7	16.33	29.00	67.33	63.67
8	17.33	30.00	73.33	69.00
9	17.33	31.33	73.33	69.00
10	15.33	33.67	70.33	66.33
11	16.00	31.33	69.33	64.33
12	16.33	32.00	71.00	66.67
13	17.00	33.00	76.00	71.33
14	18.00	28.33	72.33	67.00
15	14.67	29.67	65.33	61.67
16	15.67	31.67	68.67	65.00
17	19.00	31.00	70.00	64.67
18	17.67	30.00	69.00	66.00
19	15.33	35.00	71.33	67.00
20	16.67	31.67	72.00	66.67
Average	16.25	31.35	70.40	65.97

100, while LMC 2 detects 45 and both LMC responded to 34 signs. Performing classification on these sets of data using the GMM model, we summarized results obtained for 20 runs for 1 to 3 Gaussian mixtures in Tables 4.8, 4.9, and 4.10.

Table 4.9: Classification accuracy using 2 mixtures

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	17.00	30.67	73.67	68.67
2	18.00	32.00	75.33	70.33
3	17.67	36.67	80.00	74.33
4	17.00	34.00	74.33	70.00
5	17.67	35.67	78.00	73.33
6	19.00	35.00	80.33	76.00
7	17.33	33.67	77.33	72.33
8	17.00	33.33	75.00	69.67
9	18.00	35.67	78.33	73.67
10	18.33	35.00	79.67	74.33
11	18.33	35.33	79.00	73.67
12	16.67	31.00	73.00	68.67
13	17.00	35.00	76.33	71.33
14	17.67	33.67	75.67	70.67
15	17.00	32.00	72.67	68.33
16	16.67	34.33	76.33	71.33
17	17.67	31.67	75.00	70.33
18	16.00	33.00	75.00	70.33
19	17.00	32.67	74.67	69.33
20	17.33	35.00	76.33	71.33
Average	17.42	33.77	76.30	71.4

Table 4.10: Classification accuracy using 1 Gaussian

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	19.33	34.33	79.33	74.00
2	18.67	33.33	80.33	74.67
3	18.67	35.33	81.33	76.67
4	17.67	35.00	80.33	75.00
5	18.67	35.67	81.33	75.33
6	19.00	33.67	79.33	74.33
7	19.33	35.00	80.00	75.00
8	19.67	33.00	81.00	75.33
9	17.00	36.00	80.00	74.33
10	18.33	36.00	80.00	75.00
11	18.67	31.67	76.00	71.00
12	19.33	37.67	82.00	77.33
13	18.00	35.67	80.67	75.33
14	18.33	32.33	78.33	73.67
15	18.33	36.33	83.00	77.33
16	18.33	35.33	81.00	75.33
17	18.33	35.67	81.33	76.00
18	18.33	37.00	82.00	77.00
19	19.67	35.33	84.00	78.00
20	18.67	34.00	81.00	76.67
Average	18.67	34.93	80.63	75.32

There was no cases of rejected sign out of the 100 signs used in this case. Obtained accuracies show the advantage of combining two LMCs, since a single LMC cannot track all the one hundred signs. LMC 2, which is the LMC at the right hand side performs better in tracking the performed signs because it is close to the signer than the front LMC (i,e. LMC 1). In addition, it does well in tracking signs performed close to the signer face than LMC 1. Misclassification obtained for the presented cases of mixtures are shown in Tables 4.11, 4.12, and 4.13 respectively.

Table 4.11: Mostly misclassified signs obtained using 3 mixtures

Actual sign	Misclassification error (%)	Misclassified as	At the rate (%)
Green Peas	50	Engineer	50.00
Jeweler	30	Banana	100.00
Pilot	35	Breakfast	85.71
Muslim	45	Breakfast	44.44
Television	36	Room	71.43

Table 4.12: Mostly misclassified signs obtained using 2 mixtures

Actual sign	Misclassification error (%)	Misclassified as	At the rate (%)
Crocodile	35	Shark	57.14
Seven hundred	35	One hundred thousand	50.00
Fifty	20	Giraffe	100.00
Ten	35	One hundred thousand	85.71
Nine Hundred	40	Bedroom	50.00
Bed	40	Room	50.00
Bedspread	30	Oven	83.33
Chandelier	25	Bee	80.00
Chair	30	Kitchen	83.33
Cable	35	Air conditioner	100.00

Table 4.13: Mostly misclassified signs obtained using 1 Gaussian

Actual sign	Misclassification error (%)	Misclassified as	At the rate (%)
Tailor	25	Jewler	100.00
zucchini	40	Interpreter	62.50
One Million	30	Banana	66.67
Gorilla	35	One Million	71.43
Six hundred	30	Snake	100.00
Cock	40	Shark	62.50
Bear	40	Nullification of prayer	87.50
Hajj	35	Rubbing sock	57.14
Rubbing sock	40	Hajj	50.00
Cable	40	Air conditioner	87.5
Guest room	35	Heater	57.14

Sample image sequence for some of the misclassified signs are shown in Figures 4.14 and 4.15.

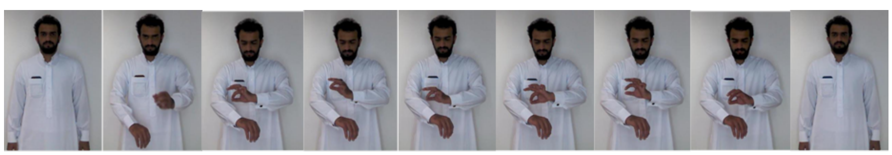


Figure 4.14: Sample image sequence for 'Tailor'

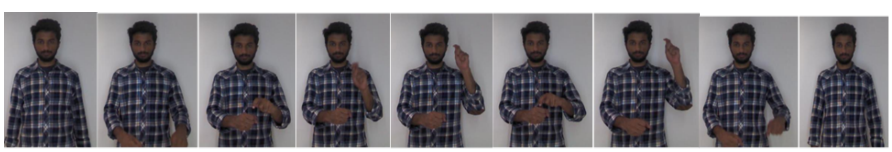


Figure 4.15: Sample image sequence for 'Jeweler'

The confusion matrices are also shown in Figures 4.16, 4.17, and 4.18.

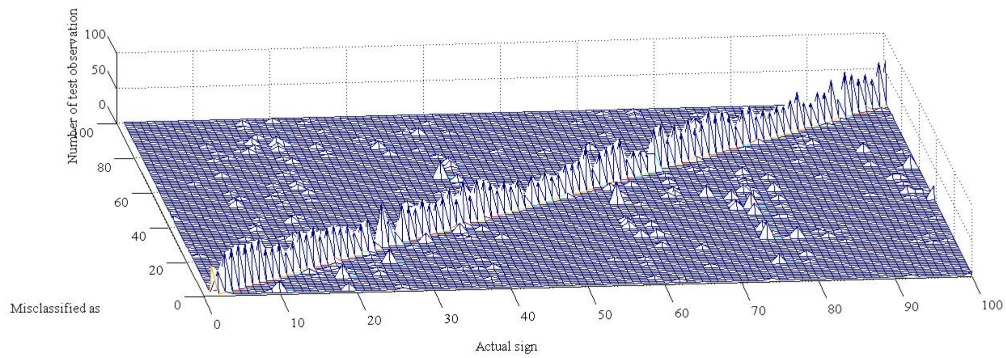


Figure 4.16: Confusion matrix for 3 mixtures

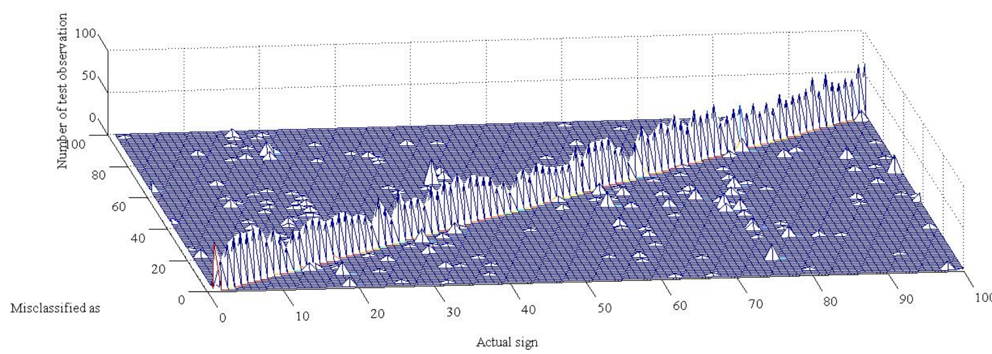


Figure 4.17: Confusion matrix for 2 mixtures

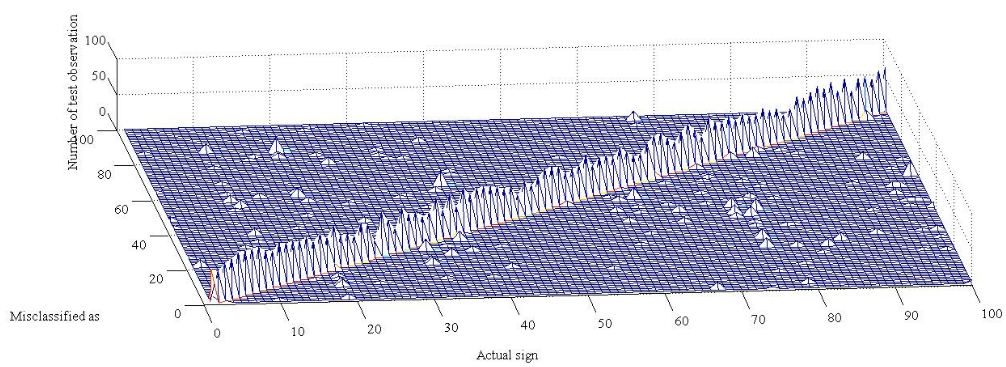


Figure 4.18: Confusion matrix for 1 mixture

Choosing another set of thresholds, we obtained the results in Figures 4.14, 4.15 and 4.16. For the case of $l_{th} = 10,\ u_{th} = 15,\ fr_{diff} = 30$, there were 15 cases of unknown signs. Similarly for $l_{th} = 4,\ u_{th} = 8,\ fr_{diff} = 25$, we have Tables 4.17, 4.18 and 4.19. For this case, we have 2 unknown cases. It should be noted that all recognition accuracies presented were computed against the total number of signs used (i.e. 100 signs).

Table 4.14: Classification accuracy using 3 mixtures

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	7.33	20.00	57.67	57.67
2	6.33	17.67	55.33	55.33
3	7.67	18.33	58.00	58.00
4	7.00	17.33	57.33	57.33
5	6.67	17.00	57.33	57.33
6	7.33	18.67	58.00	58.00
7	8.33	16.67	55.33	55.33
8	7.00	17.33	54.00	54.00
9	7.00	19.00	56.00	56.00
10	6.67	16.33	54.00	54.00
11	6.67	15.67	52.33	52.33
12	6.67	18.00	57.67	57.67
13	7.00	17.00	55.33	55.33
14	7.00	17.67	55.00	55.00
15	6.33	15.67	54.33	54.33
16	6.00	17.33	56.00	56.00
17	6.00	16.00	53.33	53.33
18	6.67	15.33	53.67	53.67
19	7.67	17.00	60.00	60.00
20	5.33	18.00	55.67	55.67
Average	6.83	17.3	55.82	55.82

Table 4.15: Classification accuracy using 2 mixtures

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	8.00	18.67	62.00	62.00
2	7.67	18.00	59.00	59.00
3	6.67	17.00	57.67	57.67
4	8.00	17.67	59.67	59.67
5	8.00	18.67	58.67	58.67
6	7.67	19.00	59.33	59.33
7	7.67	18.33	62.67	62.67
8	8.33	17.33	63.00	63.00
9	7.00	18.00	62.00	62.00
10	8.00	18.33	60.33	60.33
11	8.00	18.33	62.00	62.00
12	8.33	17.67	60.67	60.67
13	7.00	17.67	59.00	59.00
14	8.33	18.67	61.33	61.33
15	7.67	18.00	59.33	59.33
16	8.67	18.67	63.67	63.67
17	8.00	18.00	60.33	60.33
18	8.67	16.67	60.00	60.00
19	9.00	18.67	61.33	61.33
20	7.67	16.33	60.67	60.67
Average	7.92	17.98	60.63	60.63

Table 4.16: Classification accuracy using 1 Gaussian

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	8.67	20.00	65.67	65.67
2	8.00	19.33	65.00	65.00
3	9.00	19.00	65.67	65.67
4	7.67	17.00	65.67	65.67
5	8.33	16.33	62.00	62.00
6	8.33	18.33	65.67	65.67
7	7.67	18.00	62.33	62.33
8	8.00	17.33	62.67	62.67
9	8.33	19.33	65.00	65.00
10	8.00	18.00	62.00	62.00
11	7.67	19.00	65.67	65.67
12	6.67	18.00	62.00	62.00
13	7.67	16.67	62.33	62.33
14	7.67	17.33	65.00	65.00
15	7.67	22.33	70.33	70.33
16	7.67	19.67	63.33	63.33
17	7.00	16.00	61.67	61.67
18	9.00	20.00	67.00	67.00
19	8.33	19.67	64.33	64.33
20	8.00	18.33	62.00	62.00
Average	7.97	18.48	64.27	64.27

Table 4.17: Classification accuracy using 3 mixtures

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	19.00	35.33	66.67	61.67
2	17.67	37.00	67.33	61.00
3	19.67	37.00	68.33	62.00
4	22.00	34.00	67.33	62.00
5	20.00	36.67	69.00	64.00
6	20.00	36.00	69.00	63.00
7	19.00	40.67	73.33	66.33
8	19.33	37.33	70.00	64.00
9	18.33	37.67	69.67	63.33
10	19.67	39.67	74.33	67.33
11	18.33	35.00	67.67	61.33
12	19.67	39.00	70.67	64.67
13	22.00	39.00	71.33	66.00
14	20.33	37.00	69.33	63.67
15	21.33	40.00	74.00	67.67
16	19.67	33.00	64.67	59.33
17	20.67	35.67	70.00	64.00
18	18.00	36.67	65.00	60.00
19	19.00	34.67	67.00	60.67
20	18.00	38.33	63.33	63.33
Average	19.58	36.98	69.18	63.27

Table 4.18: Classification accuracy using 2 mixtures

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	21.33	40.33	75.67	68.67
2	20.00	37.33	71.66	64.67
3	19.00	40.33	73.33	66.00
4	20.33	42.33	76.00	69.00
5	20.67	39.67	71.67	66.67
6	19.67	38.67	70.33	65.33
7	21.67	39.33	75.67	69.00
8	21.00	38.67	75.33	67.33
9	21.00	38.00	74.67	66.67
10	20.00	40.00	73.33	67.67
11	21.00	38.00	71.33	65.67
12	20.33	38.00	72.00	66.33
13	21.00	41.00	77.33	70.00
14	19.67	35.33	69.67	62.00
15	21.33	39.33	75.33	68.67
16	21.67	35.33	70.33	64.67
17	19.33	36.33	69.33	63.00
18	21.33	37.00	73.00	66.67
19	22.33	39.00	75.33	69.00
20	20.00	40.33	74.00	67.67
Average	20.63	38.72	73.27	66.73

Table 4.19: Classification accuracy using 1 Gaussian

Runs	LMC 1 (%)	LMC 2 (%)	Feature level combination (%)	D-S Fusion (%)
1	22.33	43.33	81.67	73.67
2	23.67	35.33	73.33	66.00
3	22.00	40.33	78.00	70.67
4	22.33	40.67	77.67	70.67
5	21.33	41.67	77.67	71.00
6	22.33	41.67	79.33	72.33
7	23.67	40.67	79.00	72.67
8	23.33	40.67	80.00	72.33
9	22.00	42.33	79.67	72.67
10	21.67	40.67	77.33	69.33
11	24.00	39.00	78.33	70.67
12	24.33	38.67	77.00	70.00
13	23.00	41.67	80.00	72.33
14	23.33	39.00	76.67	70.33
15	22.67	38.00	76.67	69.33
16	20.67	39.00	75.00	68.00
17	23.67	43.00	82.33	74.33
18	22.67	42.00	79.33	72.67
19	24.00	43.67	81.33	75.33
20	20.67	39.67	76.00	68.67
Average	22.68	40.55	78.31	71.15

4.4 Discussion

As previously discussed, the data set used for isolated word classification consists of 10 observations per sign for 100 signs. Each of the observation contains several frames of data representing the movement of the hand sign performed by the deaf signer. Frames per observation varies depending on the length of time taken to perform the sign. Therefore, we have used the algorithm presented in Figure 4.13 to manage this difference in frames per observation. We have also assumed GMM for the data set. The optimal number of Gaussian Mixtures to be used in a data set is still an open area of research [74]. Several approaches have been suggested, such as the work of [75, 76, 74, 77]. Optimal mixtures to be used varies from one area of application to the other. In this work, we experimented with 1 to 3 mixtures. Results obtained are shown in Tables 4.8, 4.9 and 4.10 for the case of $l_{th} = 5,\ u_{th} = 10,\ fr_{diff} = 30$, numbers of frames. The best result was obtained for the case of $k = 1$. It was observed that recognition accuracy drops as the number of mixtures increases. This revealed that the data best fit for the case of a single Gaussian mixture. Similar trend was observed in all cases presented (i.e. 1-Gaussian outperform 2 and 3 mixtures of Gaussian). This was expected because most of the features histograms are unimodal except in few features like the hand roll feature shown in Figures 4.19 and 4.20.

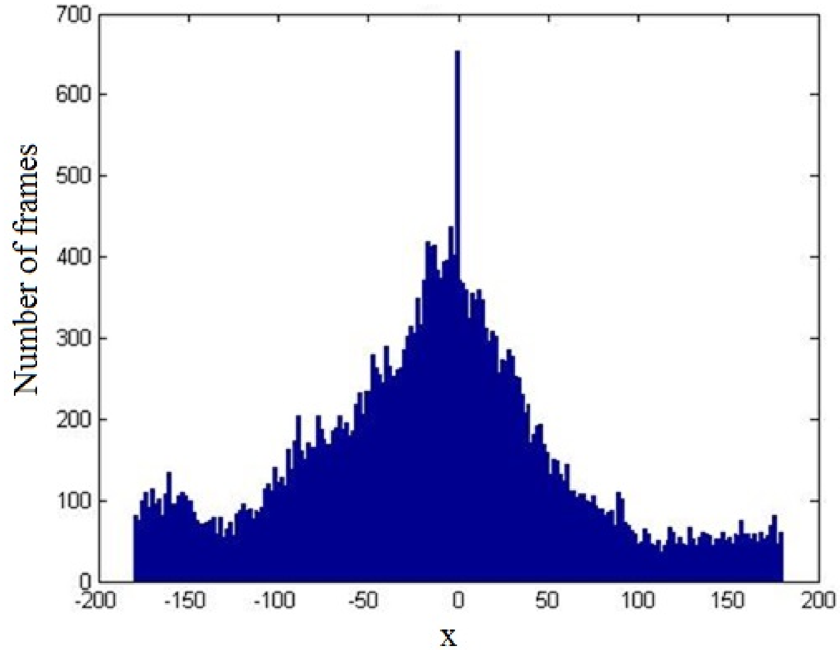


Figure 4.19: Histogram of hand roll feature for LMC 1

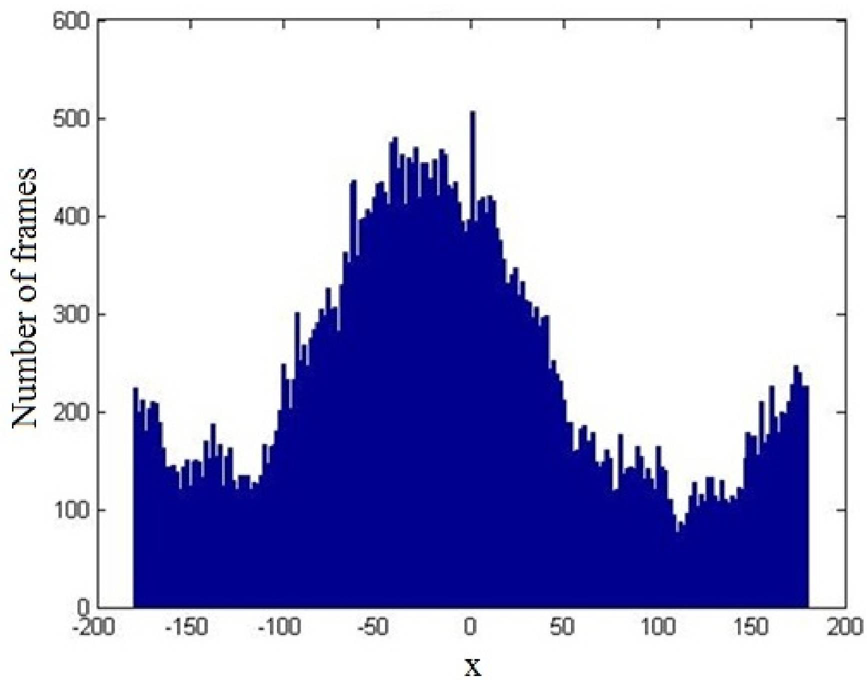


Figure 4.20: Histogram of hand roll feature for LMC 2

For convergence of GMM algorithm, we have used an error difference, between consecutive iteration, of $1x10^{-6}$ which should be achieved within a particular number of iterations. We started by using 100 iterations. We observed some cases where the algorithm fails to converge after 100 iterations for 2 and 3 mixtures of Gaussian. However, when we increased the number of iterations to 1000, the algorithm was able to converge within this number of iteration. There was no difference in the accuracies obtained for both cases of 100 and 1000 iterations, hence, the obtained results are not biased to a particular number of iterations. There is no issue of convergence with the case of 1-Gaussian and this case gave us the best classification accuracy.

We also observed the performance accuracies as we increase or decrease the thresholds related to number of frames in each observation. The threshold set during training determines the number of undecided cases. The thresholds are set according to the training data set. One major advantage of these thresholds is to increase the confidence in making the final decision. For example, if both LMCs detect the sign and LMC 1 has more data than LMC 2, exceeding a particular threshold, final decision is taken from LMC 1, etc. Increasing the threshold, increases the tendency of having more unknown cases. This, especially affects signs which take short length of time to perform. So it is better to select a minimum threshold where signs with short length will also be taken into consideration. On the other hand, making the threshold too low, signs in which the LMC responded with little number of frames are likely to be misclassified, hence reducing the recognition rate. This is obvious from results of Tables 4.10 and 4.19.

Comparing the two combination approaches used, it was generally observed that simple concatenation of the two LMCs feature vectors outperforms the use of D-S theory. The effect of the theory on the overall accuracy was not significant since it is a function of the number of times both LMC detects the sign. In addition, since features are geometric, combination of the two feature sets provide more information (it provides us with two different view of the sign performed) hence better accuracy. We also observed some cases where both combination approaches gave the same results. In this case both method do not apply because the cases of rejected signs. Hence, the results from the general case common to both are outputted.

Generally, accuracies obtained by current approaches are highly influenced by the use of color glove, constant light background settings, etc. However, results obtained in this approach, though not yet up to accuracies of current approaches, only requires the signer to perform the sign naturally. Moreover, our obtained performance accuracy still compete closely with current approaches without the need for the constrains in these approaches.

4.5 Observed Limitation of the LMC Device

During the data collection process using the LMC, the following limitation were observed:

1. Sign can only be detected within the LMC interaction box (see Figure 4.21)

which is limited in distance and less than the manufacturer specification. Consequently, some signs were completely not detected by the LMC.

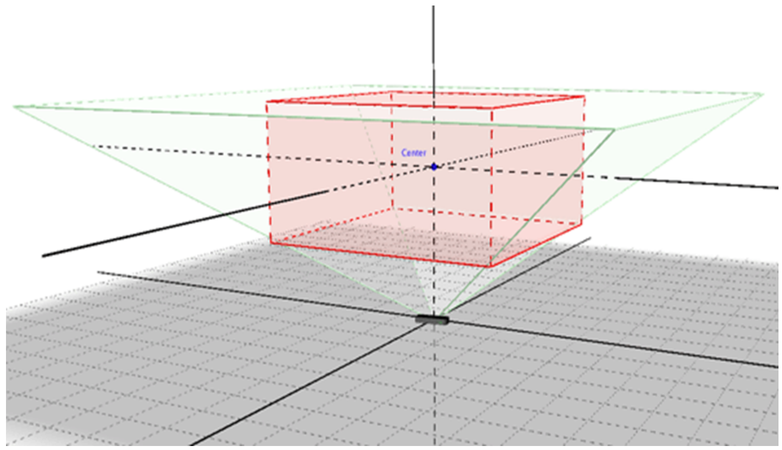


Figure 4.21: LMC interaction box

2. Signs performed beyond 25cm from the surface of the device are not detected.

3. Poor finger tracking when fingers are placed close to each other (see Figure 4.22).

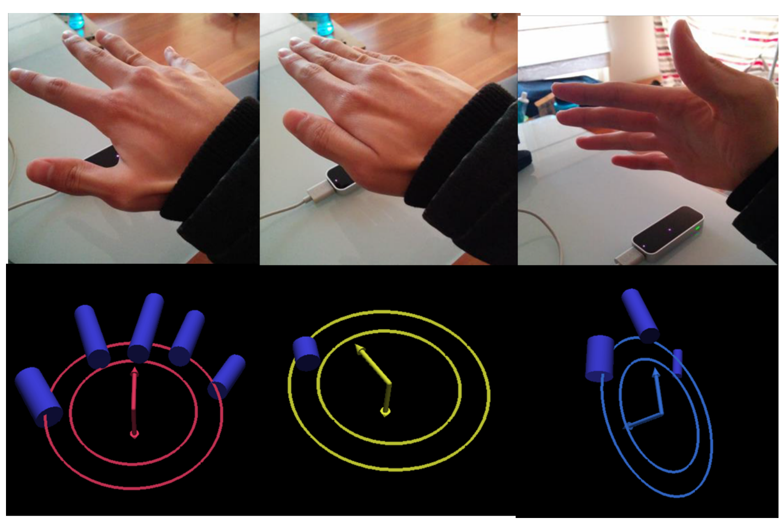


Figure 4.22: Inaccurate finger tracking when fingers are close to each other

4. Partial tracking of signs:- Some signs are not completely tracked from beginning to end of the signs, hence, making signs that are originally not similar to look similar.

4.6 Arabic sign language recognition using the Kinect

The setup for ArSLR using the Microsoft Kinect (MK) device consists of four (4) stages: the MK device, data collection, feature extraction, and classification, as shown in Figure 4.23.

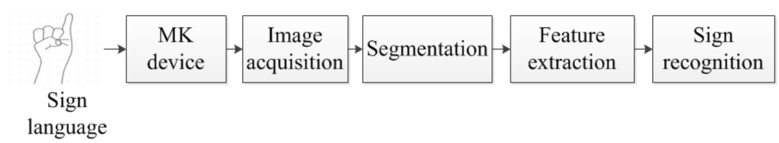


Figure 4.23: ArSLR using MK device

Brief discussion on each of these blocks in Figure 4.23 are presented below:

4.6.1 Image Acquisition using the MK device

The ArSLR setup involves the use of the recently introduced Microsoft Kinect (MK) for Windows. The MK device serves as the interface between the signer and the machine learning algorithm. The MK device sensor shares many of the core capabilities of the Kinect for Xbox 360 sensor. First, both devices contain RGB camera that stores three-channel data at a 1280 x 960 resolution at 12 frames per second or a 640 x 480 resolution at 30 frames per second. This instrument allows color images or video to be captured. Second, both devices contain an infrared (IR) emitter, which emits infrared light beams, and an IR depth sensor, which reads the IR beams reflected back to the sensor. The reflected beams are converted into depth information, measuring the distance between an object and the sensor and hence facilitating the capture of depth images. Third, both devices also contain a 4-channel microphone array for capturing sound; the microphone channels make it possible to record audio from a specific direction as well as to identify the location of the sound source and the propagation direction of the audio waves.

Finally, both devices also contain a three-axis accelerometer configured for a 2G range, where G is the acceleration due to gravity. It is possible to use the accelerometer to determine the current orientation of the sensor. The MK device also includes Near Mode, which enables the devices camera to see objects as close as 40 centimeters in front of the sensor without losing accuracy or precision, with smooth degradation out up to 3 meters [78]. The MK device is depicted in Figure 4.24 [48].

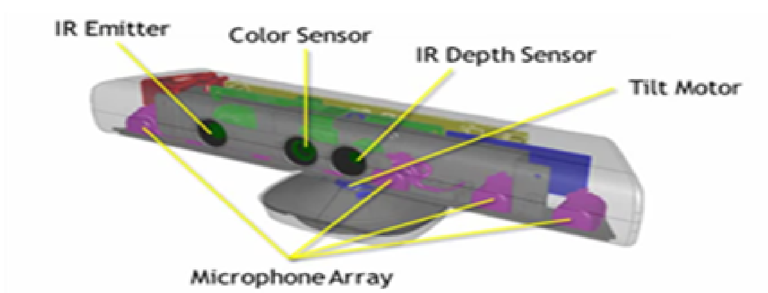


Figure 4.24: Microsoft Kinect for Windows

The interaction space of the device is the area in front of its sensors where the infrared and color sensors have an unblocked view of everything in front of the sensors. The interaction space can be increased by adjusting the built-in tilt motor. The tilt motor supports additional 27 degrees, as shown in Figure 4.25. Using this device, RGB, depth, and skeletal images as well as audio data can be acquired.

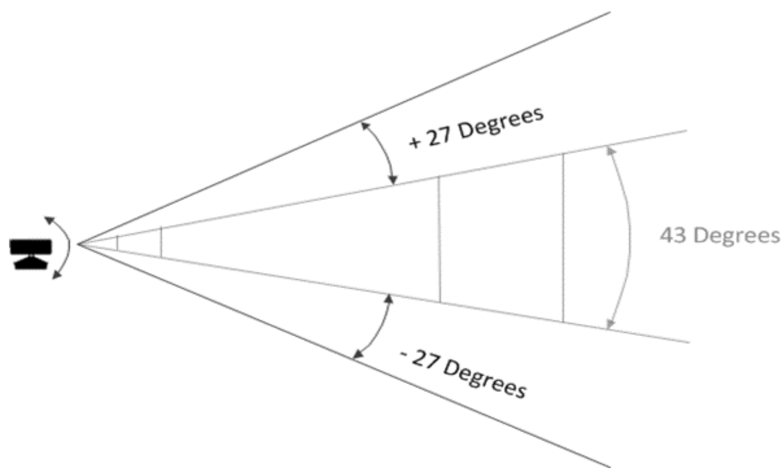


Figure 4.25: MK interaction space

As a way of comparison, we acquired RGB and videos for 40 Arabic Sign Language dynamic isolated words using the Kinect sensor which was programmed using MATLAB to capture 30 frames per seconds. Twenty samples of each letter were collected for both RGB images, giving a total of 800 samples in total. Each videos were converted back to frames to give images of sequences involved in performing each of the signs. To avoid reduce computational complexity involved in segmentation and feature extraction, samples were taken from the total frames of the video. These samples were segmented to extract the hand region. Figures 4.26show a typical frame sequence for RGB image.

Figure 4.26: Sample RGB for ArSL word 'Family'

4.6.2 Segmentation

From the sample images taken from each video, we segmented the acquired images to isolate the region of the hand representing the performed sign. This is very important especially to reduce the size of data in which the feature stage operates on. To segment a sample image, first, the Gaussian skin color model algorithm was used to extract skin portion of the image. This process leaves us with the face and hand portion of the image. Then, this image is fed to another stage which converts the image to binary image. Segmented image of Figure 4.26 is shown in Figure 4.27.

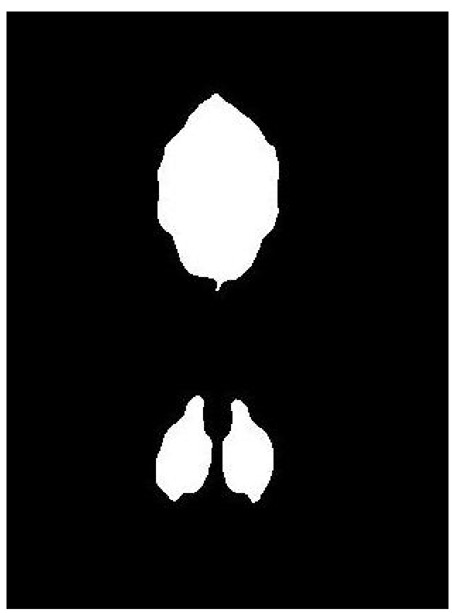


Figure 4.27: Sample segmented image of sign 'Family'

4.6.3 Feature Extraction and Classification

For feature extraction, we have used the Hu's moment as presented in [79]. This method of using the moment of the image was first developed by Hu and is often called Hu's invariant moment. The method is characterized by invariance of translation, rotation and scaling and has been successfully used in many fields. The extracted features were fed into a multilayer neural network (presented in 3) classifier.

For classification, the data was split into two 70% for training and the rest for testing. An overall classification accuracy of 81.5% was obtained with 148 cases of misclassification out of the 800 samples. The overall performance curve is shown in Figure 4.28.

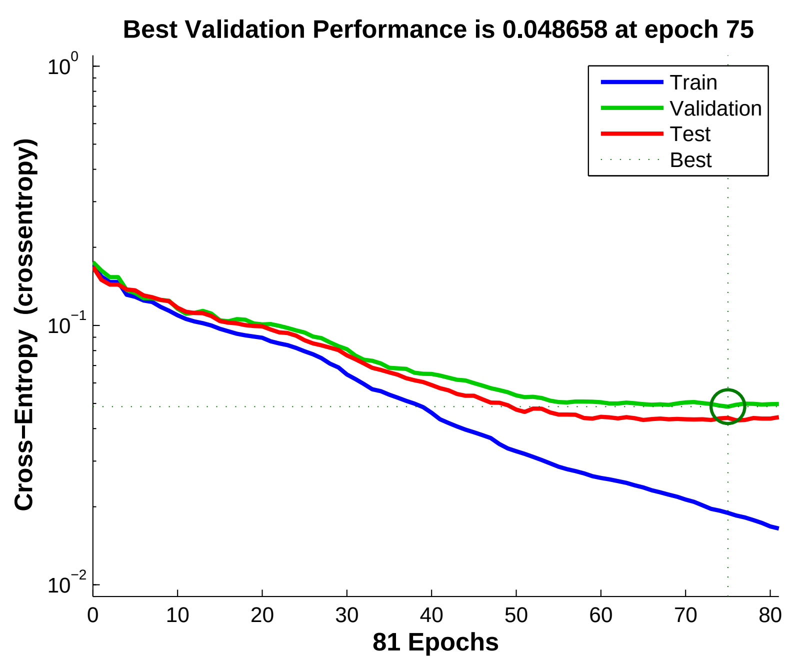


Figure 4.28: Performance curve of the network

The MK device has an advantage of being able to capture the entire body unlike the LMC device which focus on the hand region. Thus, the MK device will be able to take care of cases of signs where the LMC cannot detect. However, the performance of this approach reduces as the background lighting of the environment reduce below a certain threshold.

4.7 Summary

Results were presented for recognition of 28 ArSL letters using single LMC and two LMCs, recognition of isolated ArSL words using two LMC setup. For isolated word recognition, we have used the GMM algorithm with EM. Results were presented for a data set of 100 sign words. A maximum accuracy of 80.6% was obtained. Confusion matrices and ROC plots were also shown to visualize the performance of the classifier. As a way of comparison, we also presented ArSLR using the MK device. To sum up, we observed advantages and disadvantages in both devices. Future work will focus on having a combined setup for both devices to take advantage of combining their strength. In the next chapter, we conclude this work and give recommendation and possible future direction.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis, a new approach to Arabic Sign Language recognition system have been proposed. The approach does not require the signer to wear cumbersome glove, neither does it require specific lighting background settings. It was tested for recognition of 28 Arabic Sign Language alphabets, using single and double LMCs, and recognition of 100 isolated signs. For a single LMC on 28 alphabets, using Naive Baye's classifier (NBC) we obtained 95.5% recognition accuracy, in comparison with multilayer perceptron (MLP) which gives 94.25% accuracy. 126 instances, out of 2800, were misclassified by NBC while MLP misclassified 161 instances. For the case of two LMCs, The average accuracy (with fusion at features level) of the signs recognition using the LDA classifier was about 97.7% while the accuracy of classifier fusion using D-S theory was about 97.1%. Analysis of the misclassified signs (44 instances for feature level fusion and 80 for classifier level fusion out of 2,800 instances) revealed that most of the misclassified letter signs are similar to the signs they are classified to. On using the model for 100 isolated signs, the two-LMC setup outperforms the use of a single LMC, where LMC 1 alone gave 18.67%, LMC2 alone gave 34.93%, and from fusion of two LMCs, we have accuracy of 80.6%. We also observed that combination feature vectors from the two LMCs gave better accuracy than using D-S theory.

In conclusion, we have developed a system for Arabic Sign Language recognition. This approach, though accuracy obtained not yet up to current approaches, does not require the signer to wear cumbersome glove nor requires specific lighting background settings. Future work will concentrate on techniques to improving the obtained accuracy and increasing the database of signs.

5.2 Future Work

From observation and problems encountered in this work, several research areas for future work have been established. To achieve a robust system for Arabic Sign Language recognition using the Leap Motion Controller, observed limitations and challenges faced during the course of this thesis will need to be solved. Though the accuracy obtained from this work is without the constrained faced with current approaches, however, more work need to be done to improve the accuracy. Therefore, the following are some suggestion for future work:

1. Using the model in signer independent mode using data collected from three or more signers

2. Combination of Leap Motion Controller and Microsoft Kinect sensor

3. Investigation of other set of features from the Leap Motion device

4. Limitation with LMC detection range and tracking performance will need to be improved.

5. Future works will also include extending the model for recognition of full Arabic Sign Language sentence.

REFERENCES

[1] http://www.who.int/mediacentre/factsheets/fs300/en/.

[2] A. V. Nair *et al.*, "A review on indian sign language recognition." *International Journal of Computer Applications*, vol. 73, 2013.

[3] L. M. Company, "Api overview," https://developer.leapmotion.com/documentation/-Languages/C++/Guides/Leap_Overview.htmlid1.

[4] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. IEEE, 2014, pp. 960–965.

[5] M. Mohamed, S. Aliyu, and M. Deriche, "Prototype arabic sign language recognition using multi-sensor data fusion of two leap motion controllers," in *in The International Multi-Conference on Systems, Signals and Devices 2015, in Mahdia, Tunisia, Mar 16-19, 2015*. IEEE, 2015.

[6] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors (Basel, Switzerland)*, vol. 13, no. 5, p. 6380, 2013.

[7] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *Gesture-based communication in human-computer interaction.* Springer, 1999, pp. 103–115.

[8] M. Mohandes, "Arabic sign language recognition," in *International conference of imaging science, systems, and technology, Las Vegas, Nevada, USA*, vol. 1, 2001, pp. 753–9.

[9] O. Al-Jarrah and A. Halawani, "Recognition of gestures in arabic sign language using neuro-fuzzy systems," *Artificial Intelligence*, vol. 133, no. 1, pp. 117–138, 2001.

[10] M. Al-Rousan and M. Hussain, "Automatic recognition of arabic sign language finger spelling," *International Journal of Computers and Their Applications*, vol. 8, pp. 80–88, 2001.

[11] K. Assaleh and M. Al-Rousan, "Recognition of arabic sign language alphabet using polynomial classifiers," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 2136–2145, 2005.

[12] M. Maraqa and R. Abu-Zaiter, "Recognition of arabic sign language (arsl) using recurrent neural networks," in *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the.* IEEE, 2008, pp. 478–481.

[13] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, K. Nakamatsu *et al.*, "Arslat: Arabic sign language alphabets translator," in *Computer Information Sys-*

tems and Industrial Management Applications (CISIM), 2010 International Conference on. IEEE, 2010, pp. 590–595.

[14] E. E. Hemayed and A. S. Hassanien, "Edge-based recognizer for arabic sign language alphabet (ars2v-arabic sign to voice)," in *Computer Engineering Conference (ICENCO), 2010 International.* IEEE, 2010, pp. 121–127.

[15] S. J. Reyadh Naoum, Hussein H. Owaied, "Development of a new arabic sign language recognition using k-nearest neighbor algorithm," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, pp. 1173–1178, 2012.

[16] I. M. Al-Saihati, *Real time Arabic sign language recognition.* ProQuest, 2006.

[17] M. Mohandes and M. Deriche, "Image based arabic sign language recognition," in *Signal Processing and Its Applications, 2005. Proceedings of the Eighth International Symposium on*, vol. 1. IEEE, 2005, pp. 86–89.

[18] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, "A signer-independent arabic sign language recognition system using face detection, geometric features, and a hidden markov model," *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 422–433, 2012.

[19] T. Shanableh and A. Khaled, "Arabic sign language recognition in user-independent mode," in *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on.* IEEE, 2007, pp. 597–600.

[20] T. Shanableh and K. Assaleh, "Two tier feature extractions for recognition of isolated arabic sign language using fisher's linear discriminants," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2. IEEE, 2007, pp. 501–504.

[21] S. Tamer and K. Assaleh, "Video-based feature extraction techniques for isolated arabic sign language recognition," in *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on.* IEEE, 2007, pp. 1–4.

[22] A. A. Youssif, A. E. Aboutabl, and H. H. Ali, "Arabic sign language (arsl) recognition system using hmm," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 2, no. 11, 2011.

[23] M. M. Zaki and S. I. Shaheen, "Sign language recognition using a combination of new vision based features," *Pattern Recognition Letters*, vol. 32, no. 4, pp. 572–577, 2011.

[24] A. Samir and M. Aboul-Ela, "Error detection and correction approach for arabic sign language recognition," in *Computer Engineering & Systems (ICCES), 2012 Seventh International Conference on.* IEEE, 2012, pp. 117–123.

[25] A. S. Elons, M. Abull-Ela, and M. F. Tolba, "A proposed pcnn features quality optimization technique for pose-invariant 3d arabic sign language recognition," *Applied Soft Computing*, vol. 13, no. 4, pp. 1646–1660, 2013.

[26] A. Elons, "Gpu implementation for arabic sign language real time recognition using multi-level multiplicative neural networks," in *Computer Engineering & Systems (ICCES), 2014 9th International Conference on.* IEEE, 2014, pp. 360–367.

[27] M. Al-Rousan, O. Al-Jarrah, and M. Al-Hammouri, "Recognition of dynamic gestures in arabic sign language using two stages hierarchical scheme," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 14, no. 3, pp. 139–152, 2010.

[28] S. M. Halawani, "Arabic sign language translation system on mobile devices," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 1, pp. 251–256, 2008.

[29] 5DT, "5dt products." http://www.5dt.com/products/pdataglovemri.html.

[30] Warepin, "Wii power glove." http://www.warepin.com/wii-power-gloves/.

[31] C. Systems, "Cyberglove," http://www.cyberglovesystems.com/products/cyberglove-ii/photos-video.

[32] M. Mohandes and S. Buraiky, "Automation of the arabic sign language recognition using the powerglove," *AIML Journal*, vol. 7, no. 1, pp. 41–46, 2007.

[33] M. Mohandes, T. Halawani, S. Al-Baiyat *et al.*, "Automation of the arabic sign language recognition," in *Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on.* IEEE, 2004, pp. 479–480.

[34] K. Assaleh, T. Shanableh, and M. Zourob, "Low complexity classification system for glove-based arabic sign language recognition," in *Neural Information Processing*. Springer, 2012, pp. 262–268.

[35] T. Ritchings, A. Khadragi, and M. Saeb, "An intelligent computer-based system for sign language tutoring," *Assistive Technology*, vol. 24, no. 4, pp. 299–308, 2012.

[36] M. A. Mohandes, "Recognition of two-handed arabic signs using the cyberglove," *Arabian Journal for Science and Engineering*, vol. 38, no. 3, pp. 669–677, 2013.

[37] M. Mohandes and M. Deriche, "Arabic sign language recognition by decisions fusion using dempster-shafer theory of evidence," in *Computing, Communications and IT Applications Conference (ComComAp), 2013*. IEEE, 2013, pp. 90–94.

[38] A. Elons, M. Ahmed, H. Shedid, and M. Tolba, "Arabic sign language recognition using leap motion sensor," in *Computer Engineering & Systems (ICCES), 2014 9th International Conference on*. IEEE, 2014, pp. 368–373.

[39] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. ACM, 2013, pp. 175–178.

[40] O. Kainz, F. Jakab *et al.*, "Approach to hand tracking and gesture recognition based on depth-sensing cameras and emg monitoring," *Acta Informatica Pragensia*, vol. 3, no. 1, pp. 104–112, 2014.

[41] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in *Image Processing (ICIP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 1565–1569.

[42] G. Marin, F. Dominio, and Z. Pietro, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools and Applications*, pp. 1–25, 2015.

[43] V. B. R. S. T. P. Karthick, N. Prathiba, "Transforming indian sign language into text using leap motion," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 4, pp. 10 906–10 910, 2014.

[44] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou, "Sign language recognition and translation with kinect," in *IEEE Conf. on AFGR*, 2013.

[45] A. Agarwal and M. K. Thakur, "Sign language recognition using microsoft kinect," in *Contemporary Computing (IC3), 2013 Sixth International Conference on.* IEEE, 2013, pp. 181–185.

[46] M. Geetha, C. Manjusha, P. Unnikrishnan, and R. Harikrishnan, "A vision based dynamic gesture recognition of indian sign language on kinect based depth images," in *Emerging Trends in Communication, Control, Signal Pro-*

cessing & Computing Applications (C2SPCA), 2013 International Conference on. IEEE, 2013, pp. 1–7.

[47] A. Memis and S. Albayrak, "Turkish sign language recognition using spatio-temporal features on kinect rgb video sequences and depth maps," in *Signal Processing and Communications Applications Conference (SIU), 2013 21st.* IEEE, 2013, pp. 1–4.

[48] C. Waithayanon and C. Aporntewan, "A motion classifier for microsoft kinect," in *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on.* IEEE, 2011, pp. 727–731.

[49] A. Tuntakurn, S. S. Thongvigitmanee, V. Sa-Ing, S. S. Makhanov, and S. Hasegawa, "Natural interaction on 3d medical image viewer software," in *Biomedical Engineering International Conference (BMEiCON), 2012.* IEEE, 2012, pp. 1–5.

[50] J. Venugopalan, C. Cheng, T. H. Stokes, and M. D. Wang, "Kinect-based rehabilitation system for patients with traumatic brain injury," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE.* IEEE, 2013, pp. 4625–4628.

[51] E. E. Stone, M. Butler, A. McRuer, A. Gray, J. Marks, and M. Skubic, "Evaluation of the microsoft kinect for screening acl injury," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE.* IEEE, 2013, pp. 4152–4155.

[52] R. A. El-laithy, J. Huang, and M. Yeh, "Study on the use of microsoft kinect for robotics applications," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*. IEEE, 2012, pp. 1280–1288.

[53] F. Soltani, F. Eskandari, and S. Golestan, "Developing a gesture-based game for deaf/mute people using microsoft kinect," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on*. IEEE, 2012, pp. 491–495.

[54] J. Kittler, "A framework for classifier fusion: Is it still needed?" in *Advances in Pattern Recognition*. Springer, 2000, pp. 45–56.

[55] M. D. Muhlbaier, A. Topalis, and R. Polikar, "Learn. nc: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes," *Neural Networks, IEEE Transactions on*, vol. 20, no. 1, pp. 152–168, 2009.

[56] J. Kittler, "Combining classifiers: A theoretical framework," *Pattern analysis and Applications*, vol. 1, no. 1, pp. 18–27, 1998.

[57] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.

[58] M. A. A. Siddiqui, "Fusion of ecg/eeg for improved automatic seizure detection using dempster shafer theory of evidence," p. 143, 2011, copyright - Copyright ProQuest, UMI Dissertations Publishing 2011;

Last updated - 2014-01-08; First page - n/a. [Online]. Available: http://search.proquest.com/docview/1313565819?accountid=27795

[59] I. Naseem, "Combining classifiers using the dempster-shafer theory of evidence," Ph.D. dissertation, Citeseer, 2005.

[60] Y. Zhan, H. Leung, K.-C. Kwak, and H. Yoon, "Automated speaker recognition for home service robots using genetic algorithm and dempster–shafer fusion technique," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 9, pp. 3058–3068, 2009.

[61] A. Lindner, "The leap motion controller and touchless technology," http://dartmouthbusinessjournal.com/2013/08/the-leap-motion-controller-and-touchless-technology.

[62] "Leap motion forum," https://forums.leapmotion.com.

[63] D. E. Rumelhart, J. L. McClelland, P. R. Group *et al.*, *Parallel distributed processing.* IEEE, 1988, vol. 1.

[64] P. D. Wasserman, *Advanced methods in neural computing.* John Wiley & Sons, Inc., 1993.

[65] R. P. Lippmann, "An introduction to computing with neural nets," *ASSP Magazine, IEEE*, vol. 4, no. 2, pp. 4–22, 1987.

[66] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification.* John Wiley & Sons, 2012.

[67] S. Theodoridis and K. Koutroumbas, “Chapter 2 - classifiers based on bayes decision theory,” in *Pattern Recognition (Fourth Edition)*, fourth edition ed., S. Theodoridis and K. Koutroumbas, Eds. Boston: Academic Press, 2009, pp. 13 – 89. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781597492720500049

[68] D. Harmanec, “Toward a characterization of uncertainty measure for the dempster-shafer theory,” in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 255–261.

[69] G. McLachlan, *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, 2004, vol. 544.

[70] D. Reynolds, “Gaussian mixture models,” in *Encyclopedia of Biometrics*. Springer, 2009, pp. 659–663.

[71] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[72] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[73] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[74] K. Yu, "Generating gaussian mixture models by model selection for speech recognition."

[75] Y. Lee, K. Y. Lee, and J. Lee, "The estimating optimal number of gaussian mixtures based on incremental k-means for speaker identification," *International Journal of Information Technology*, vol. 12, no. 7, pp. 13–21, 2006.

[76] P. McKenzie and M. Alder, "Selecting the optimal number of components for a gaussian mixture model," in *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*. IEEE, 1994, p. 393.

[77] H. Tenmoto, M. Kudo, and M. Shimbo, "Mdl-based selection of the number of components in mixture models for pattern classification," in *Advances in Pattern Recognition*. Springer, 1998, pp. 831–836.

[78] Microsoft, "Kinect for windows," http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx.

[79] W. Hongliang, Y. Xiao, and D. Haifei, "Method of features extraction for infrared image recognition based on image moment," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, vol. 9, 2010.

Appendix: Images describing features

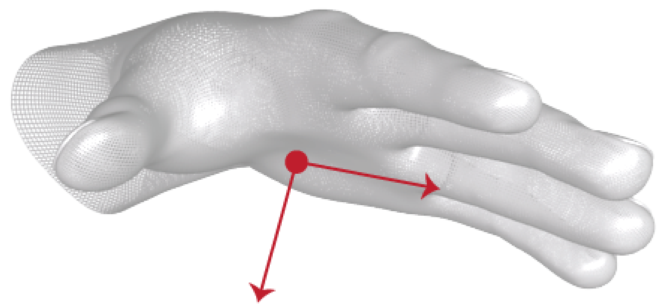


Figure A.1: Hand palm posiion

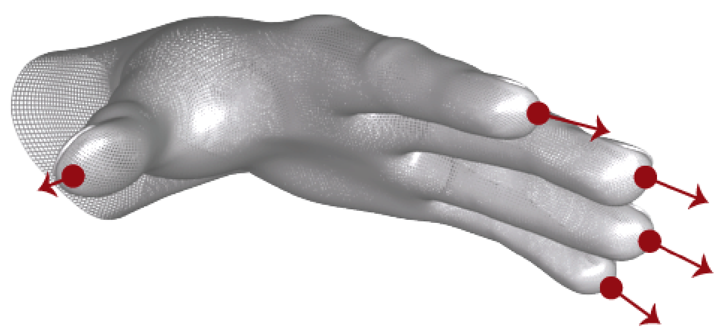


Figure A.2: Hand tip position

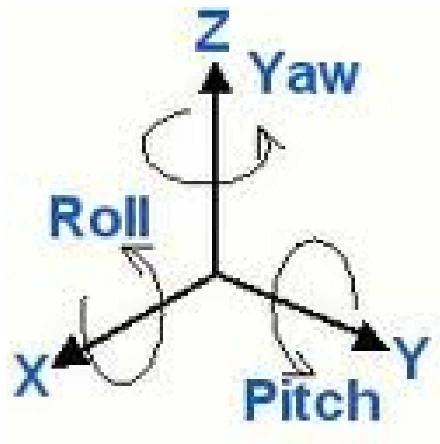


Figure A.3: Hand pitch, yaw and roll

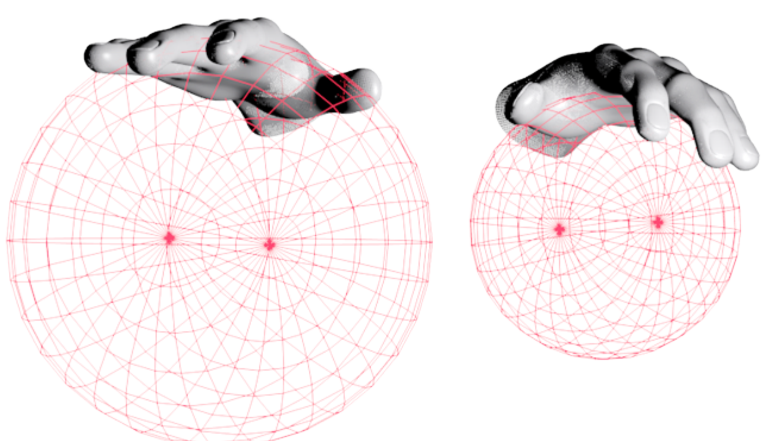


Figure A.4: Hand sphere radius

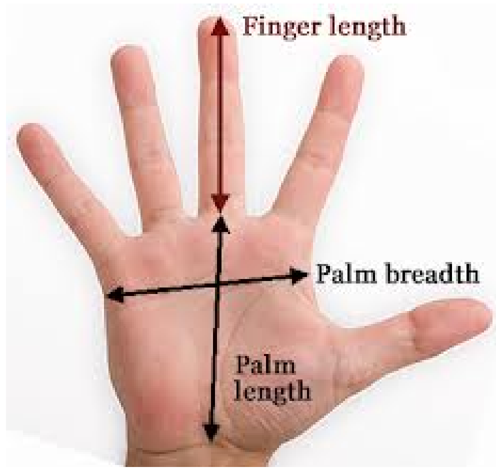


Figure A.5: Finger length

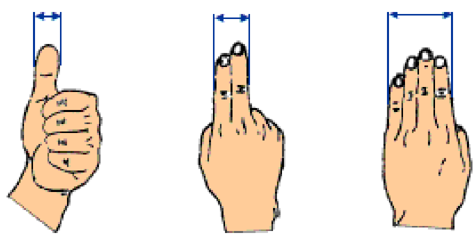


Figure A.6: Average finger width

ALIYU Salihu Oladimeji

CONTACTS: Telephone Number - **+2347039335074, +966553264133**
salihu.aliyu@futminna.edu.ng.
Department of Telecommunication Engineering, Federal University of Technology, Minna, Nigeria

PERSONAL DATA:

Date of birth - 21 OCT 1982
Sex - Male
Origin - Ilorin-West, Nigeria
Marital status - Married
Profession – Electrical and Computer Engineering
Spoken Languages - English, Arabic, Yoruba, Hausa and Ebira

EDUCATION AND QUALIFICATION

2013-2015 King Fahd University of Petroleum and Minerals, Dharan, Saudi Arabia
Master of Science, Electrical Engineering.

2005-2010 Federal University of Technology, Minna, Niger State B.Eng
(Elect. And Compt. Engr.)
First Class Honor.

2004-2005 College of Arabic and Islamic Studies, Ilorin, Nigeria. Diploma in Arabic and Islamic Studies.
Distinction.

1998-2004 Ansarul-Islam Grammar School Ijomu-oro Kwara State
Senior Secondary School Certificate.

1995-1998 Staff Comprehensive Sec. School Ajaokuta
Junior Secondary School Certificate

1990-1995 Camp Staff School Ajaokuta, Kogi State
First School leaving Certificate

EXPERIENCE GAINED AFTER FIRST DEGREE

Period	**Experience**
13/07/12 – Till date	Graduate Assistant, Federal University of Technology, Minna, Niger State, Nigeria.
05/07/11 – 04/07/12	National Youth Service Corps (NYSC), Physics Teacher, Community Secondary School, Agulu-Awka, Anambra State, Nigeria.

01/12/10 – 02/07/11 Electronics Circuit Design Engineer at ManHenrycee Electronics Construction Company, Ilorin Nigeria.

UNDERGRADUATE PRACTICAL TRAININGS

Period	Experience
10/01/08 – 30/10/08	Certificate of Proficiency in Computer Fundamentals, Internet Applications, Windows XP, Microsoft Office Word etc from AFRIHUB NIG. LTD
04/05/09 – 06/11/09	Industrial Training, Manhenrycee Electronics Construction Company Limited, Ilorin, Nigeria
05/08/09 –10/01/10	General Engineering Data services, SKANNET Internet Services Providers, Ilorin, Nigeria.
15/01/10- Till date	Embedded System Design using 8051 Microcontrollers and Microchip

CONFERENCES:

1. **S. Aliyu**, "A Prototype Arabic Sign Language Recognition System using Fusion of Leap Motion Controller and Microsoft Kinect (Innovation Track)", The Sixth Scientific Conference for Students of Higher Education in K.S.A. March 30 - April 02, 2015, Venue: Jeddah, KSA.

2. M. Mohandes, N. Tassaduq, **S. Aliyu**, and M. Deriche, "*A Preference Based Smart Parking System: KFUPM Case Study*" in The 31st International Review of Progress in Applied Computational Electromagnetics (ACES 2015), in Williamsburg, Virginia, USA, Mar 22-26, 2015.

3. M. Mohandes, **S. Aliyu**, and M. Deriche, "*Prototype Arabic Sign Language Recognition using Multi-Sensor Data Fusion of Two Leap Motion Controllers*", in The International Multi-Conference on Systems, Signals and Devices 2015 in Mahdia, Tunisia, Mar 16-19, 2015.

4. M. Mohandes, **S. Aliyu**, and M. Deriche, "*Arabic sign language recognition using the leap motion controller*," in Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on. IEEE, 2014, Istanbul, Turkey, June 1-4, 2014, pp. 960–965.

5. A.K. Agboola, O.M. Olaniyi, **S. Aliyu**, "Agriculture and National Transformation Agenda: Faculty Staff Collaborative Invention of Models For Bird-Egg Incubator" ISTEAMS Research NEXUS Conference ABUAD 2014, International Conference on Science, Technology, Education, Arts, Management & the Social Sciences - Afe Babalola University Ado Ekiti, Nigeria. May 29 - 31, 2014.

6. A.K. Agboola, O.M. Olaniyi, **S. Aliyu** and B.A. Ayanwale, "Agricultural Development in Niger State: What Role can Resident Universities Play?", ISTEAMS Research NEXUS Conference ABUAD 2014, International Conference on Science, Technology, Education, Arts, Management & the Social Sciences - Afe Babalola University Ado Ekiti, Nigeria. May 29 - 31, 2014.

7. Agboola, A. K., Olaniyi, O.M., **Aliyu S.O**., Ayanwale, B.A:"Increasing Livestock Production in Nigeria: Development of Cost-Effective Models for Bird-Egg Incubator" presented at Second International Conference on Engineering and Technology Research (FET CONFERENCE 2013) held at Ladoke Akintola University of Technology, Ogbomosho, Nigeria, from 26th to 28th March, 2013.

8. Oladeji A.S, B. F. Sule, **Aliyu S.O** and A. Abdulkarim: "The Development Of Wind-Solar-Hydro Energy Generating Systems in Nigeria" presented at PTDF_UMYU International Conference on Renewable Energy, Katsina, 2012, held at the Auditorium, Umaru Musa Yar'adua Univeristy, Katsina from 3rd to 5th Sept 2012.

PUBLICATIONS:

Agboola, A. K., Olaniyi, O.M., **Aliyu S.O**., Ayanwale, B.A: "*Increasing Livestock Production in Nigeria: Development of Cost-Effective Models for Bird-Egg Incubator*", International Journal of Emerging Technology and Advanced Engineering, Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013).

PATENT:

M. Mohandes, M. Deriche, and S. Aliyu, "ARABIC SIGN LANGUAGE RECOGNITION USING MULTI-SENSOR DATA FUSION" US Patent, Application Number: 62/113,276.

Research Projects:

"Arabic Sign Language Recognition using Multi-Sensor Data Fusion of Leap Motion Controllers" Funded by DSR-Fast Track, PI: Dr. M. Mohandes. Status: *In Progress*

"Arabic Sign Language Recognition Using the Microsoft Kinect Device and the Dempster-Shafer Theory of Evidence" Submitted to NSTIP. PI: Dr. M. Mohandes.

Capstone Projects Supervised:

EE411-Arabic Sign Language Recognition System using the Microsoft Kinect – 2015

EE411- Smart Irrigation System - 2015

EE411- Arabic Sign Language Recognition system using the Leap Motion Controller - 2014

AWARDS:

- ELF Petroleum Nigeria Limited.
  EPNL Merit Scholarship 2006/2007.
- Dean List Award, 2006, Federal University of Technology, Minna Niger State, Nigeria.

PERSONAL SKILLS:

Good in Basic Programming, C/C++, Assembly Language, MATLAB
Good in Electronics Circuit Design and Construction
Great communication skills
Computer literate.

HOBBIES:

Reading, Swimming and Playing Football

REFEREES

Dr. Mohamed Mohandes, Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

Dr. Mohamed Deriche, Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

Dr. Zerguine, Azzedine, Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

Publications

- **Patent:** M. Mohandes, M. Deriche, and **S. Aliyu**, "ARABIC SIGN LANGUAGE RECOGNITION USING MULTI-SENSOR DATA FUSION" US Patent, Application Number: 62/113,276.-**Filled**.

- M. Mohandes, **S. Aliyu**, and M. Deriche, "Prototype Arabic Sign Language Recognition using Multi-Sensor Data Fusion of Two Leap Motion Controllers", in The International Multi-Conference on Systems, Signals and Devices 2015 in Mahdia, Tunisia, Mar 16-19, 2015.

- M. Mohandes, **S. Aliyu**, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on. IEEE, 2014, Istanbul, Turkey, June 1-4, 2014, pp. 960–965.